

Transactions on Computational Science
and Computational Intelligence

Hamid R. Arabnia
Leonidas Deligiannidis
Fernando G. Tinetti
Quoc-Nam Tran *Editors*

Advances in Software Engineering, Education, and e-Learning

Proceedings from FECS'20, FCS'20,
SERP'20, and EEE'20

 Springer

Transactions on Computational Science and Computational Intelligence

Series Editor

Hamid Arabnia

Department of Computer Science

The University of Georgia

Athens, GA

USA

Computational Science (CS) and Computational Intelligence (CI) both share the same objective: finding solutions to difficult problems. However, the methods to the solutions are different. The main objective of this book series, “Transactions on Computational Science and Computational Intelligence”, is to facilitate increased opportunities for cross-fertilization across CS and CI. This book series publishes monographs, professional books, contributed volumes, and textbooks in Computational Science and Computational Intelligence. Book proposals are solicited for consideration in all topics in CS and CI including, but not limited to, Pattern recognition applications; Machine vision; Brain-machine interface; Embodied robotics; Biometrics; Computational biology; Bioinformatics; Image and signal processing; Information mining and forecasting; Sensor networks; Information processing; Internet and multimedia; DNA computing; Machine learning applications; Multi-agent systems applications; Telecommunications; Transportation systems; Intrusion detection and fault diagnosis; Game technologies; Material sciences; Space, weather, climate systems, and global changes; Computational ocean and earth sciences; Combustion system simulation; Computational chemistry and biochemistry; Computational physics; Medical applications; Transportation systems and simulations; Structural engineering; Computational electro-magnetic; Computer graphics and multimedia; Face recognition; Semiconductor technology, electronic circuits, and system design; Dynamic systems; Computational finance; Information mining and applications; Biometric modeling; Computational journalism; Geographical Information Systems (GIS) and remote sensing; Ubiquitous computing; Virtual reality; Agent-based modeling; Computational psychometrics; Affective computing; Computational economics; Computational statistics; and Emerging applications.

For further information, please contact Mary James, Senior Editor, Springer, mary.james@springer.com.

More information about this series at <http://www.springer.com/series/11769>

Hamid R. Arabnia • Leonidas Deligiannidis
Fernando G. Tinetti • Quoc-Nam Tran
Editors

Advances in Software Engineering, Education, and e-Learning

Proceedings from FECS'20, FCS'20,
SERP'20, and EEE'20

 Springer

Editors

Hamid R. Arabnia
Department of Computer Science
University of Georgia
Athens, GA, USA

Leonidas Deligiannidis
School of Computing and Data Sciences
Wentworth Institute of Technology
Boston, MA, USA

Fernando G. Tinetti
Facultad de Informática - CIC PBA
National University of La Plata
La Plata, Argentina

Quoc-Nam Tran
Department of Computer Science
Southeastern Louisiana University
Hammond, LA, USA

ISSN 2569-7072

ISSN 2569-7080 (electronic)

Transactions on Computational Science and Computational Intelligence

ISBN 978-3-030-70872-6

ISBN 978-3-030-70873-3 (eBook)

<https://doi.org/10.1007/978-3-030-70873-3>

© Springer Nature Switzerland AG 2021

This work is subject to copyright. All rights are reserved by the Publisher, whether the whole or part of the material is concerned, specifically the rights of translation, reprinting, reuse of illustrations, recitation, broadcasting, reproduction on microfilms or in any other physical way, and transmission or information storage and retrieval, electronic adaptation, computer software, or by similar or dissimilar methodology now known or hereafter developed.

The use of general descriptive names, registered names, trademarks, service marks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

The publisher, the authors, and the editors are safe to assume that the advice and information in this book are believed to be true and accurate at the date of publication. Neither the publisher nor the authors or the editors give a warranty, expressed or implied, with respect to the material contained herein or for any errors or omissions that may have been made. The publisher remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

This Springer imprint is published by the registered company Springer Nature Switzerland AG

The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

It gives us great pleasure to introduce this collection of papers that were presented at the following international conferences: Scientific Computing (CSC 2020); Parallel & Distributed Processing Techniques and Applications (PDPTA 2020); Modeling, Simulation & Visualization Methods (MSV 2020); and Grid, Cloud, & Cluster Computing (GCC 2020). These four conferences were held simultaneously (same location and dates) at Luxor Hotel (MGM Resorts International), Las Vegas, USA, July 27–30, 2020. This international event was held using a hybrid approach, that is, “in-person” and “virtual/online” presentations and discussions.

This book is composed of ten Parts. Parts I through IV (composed of 27 chapters) include articles that address various challenges in the area of scientific computing (CSC). Parts V through VII (composed of 31 chapters) include articles that discuss advances in the area of parallel and distributed processing (PDPTA). Recent progress in the fields of modeling, simulation, and visualization methods (MSV) appear in Parts VIII through IX (composed of 17 chapters). Lastly, Part V (composed of 10 chapters) presents advances in grid, cloud, and cluster computing (GCC).

An important mission of the World Congress in Computer Science, Computer Engineering, and Applied Computing, CSCE (a federated congress to which this event is affiliated with), includes “*Providing a unique platform for a diverse community of constituents composed of scholars, researchers, developers, educators, and practitioners. The Congress makes concerted effort to reach out to participants affiliated with diverse entities (such as: universities, institutions, corporations, government agencies, and research centers/labs) from all over the world. The congress also attempts to connect participants from institutions that have **teaching** as their main mission with those who are affiliated with institutions that have **research** as their main mission. The congress uses a quota system to achieve its institution and geography diversity objectives.*” By any definition of diversity, this congress is among the most diverse scientific meeting in the USA. We are proud to report that this federated congress had authors and participants from 54 different

nations representing variety of personal and scientific experiences that arise from differences in culture and values.

The program committees (refer to subsequent pages for the list of the members of committees) would like to thank all those who submitted papers for consideration. About 50% of the submissions were from outside the USA. Each submitted paper was peer reviewed by two experts in the field for originality, significance, clarity, impact, and soundness. In cases of contradictory recommendations, a member of the conference program committee was charged to make the final decision; often, this involved seeking help from additional referees. In addition, papers whose authors included a member of the conference program committee were evaluated using the double-blind review process. One exception to the above evaluation process was for papers that were submitted directly to chairs/organizers of pre-approved sessions/workshops; in these cases, the chairs/organizers were responsible for the evaluation of such submissions. The overall paper acceptance rate for regular papers was 20%; 18% of the remaining papers were accepted as short and/or poster papers.

We are grateful to the many colleagues who offered their services in preparing this book. In particular, we would like to thank the members of the Program Committees of individual research tracks as well as the members of the Steering Committees of CSC 2020, PDPTA 2020, MSV 2020, and GCC 2020; their names appear in the subsequent pages. We would also like to extend our appreciation to over 500 referees.

As sponsors-at-large, partners, and/or organizers, each of the followings (separated by semicolons) provided help for at least one research track: Computer Science Research, Education, and Applications (CSREA); US Chapter of World Academy of Science; American Council on Science and Education & Federated Research council; and Colorado Engineering Inc. In addition, a number of university faculty members and their staff, several publishers of computer science and computer engineering books and journals, chapters and/or task forces of computer science associations/organizations from three regions, and developers of high-performance machines and systems provided significant help in organizing the event as well as providing some resources. We are grateful to them all.

We express our gratitude to all authors of the articles published in this book and the speakers who delivered their research results at the congress. We would also like to thank the followings: UCMSS (Universal Conference Management Systems & Support, California, USA) for managing all aspects of the conference; Dr. Tim Field of APC for coordinating and managing the printing of the programs; the staff of Luxor Hotel (MGM Convention) for the professional service they provided; and Ashu M. G. Solo for his help in publicizing the congress. Last but not least, we would like to thank Ms. Mary James (Springer Senior Editor in New York) and Arun Pandian KJ (Springer Production Editor) for the excellent professional service they provided for this book project.

Hamid R. Arabnia, Leonidas Deligiannidis, Fernando G. Tinetti, Quoc-Nam Tran, Ray Hashemi, Azita Bahrami

Book Co-Editors, Chapter Co-Editors & Vice-Chairs: FECS 2020, FCS 2020, SERP 2020, EEE 2020

Athens, GA, USA

Boston, MA, USA

La Plata, Argentina

Hammond, LA, USA

Hamid R. Arabnia

Leonidas Deligiannidis

Fernando G. Tinetti

Quoc-Nam Tran

Frontiers in Education: Computer Science & Computer Engineering

FECS 2020 – Program Committee

- *Prof. Afrand Agah; Department of Computer Science, West Chester University of Pennsylvania, West Chester, PA, USA*
- *Prof. Abbas M. Al-Bakry (Congress Steering Committee); University President, University of IT and Communications, Baghdad, Iraq*
- *Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Prof. Mehran Asadi; Department of Business and Entrepreneurial Studies, The Lincoln University, Pennsylvania, USA*
- *Dr. Azita Bahrami (Vice-Chair); President, IT Consult, USA*
- *Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de Valencia (UPV), Department of Computer Engineering (DISCA), Valencia, Spain*
- *Prof. Juan Jose Martinez Castillo; Director, The Acantelys Alan Turing Nikola Tesla Research Group and GIPEB, Universidad Nacional Abierta, Venezuela*
- *Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Zhangisina Gulnur Davletzhanovna; Vice-rector of the Science, Central-Asian University, Kazakhstan, Almaty, Republic of Kazakhstan; Vice President of International Academy of Informatization, Kazakhstan, Almaty, Republic of Kazakhstan*
- *Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*

- *Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA*
- *Prof. George A. Gravvanis (Congress Steering Committee); Director, Physics Laboratory & Head of Advanced Scientific Computing, Applied Math & Applications Research Group; Professor of Applied Mathematics and Numerical Computing and Department of ECE, School of Engineering, Democritus University of Thrace, Xanthi, Greece*
- *Prof. Ray Hashemi (Vice-Chair); College of Engineering and Computing, Georgia Southern University, Georgia, USA*
- *Prof. Houcine Hassan; Department of Computer Engineering (Systems Data Processing and Computers), Universitat Politecnica de Valencia, Spain*
- *Prof. George Jandieri (Congress Steering Committee); Georgian Technical University, Tbilisi, Georgia; Chief Scientist, The Institute of Cybernetics, Georgian Academy of Science, Georgia; Ed. Member, International Journal of Microwaves and Optical Technology, The Open Atmospheric Science Journal, American Journal of Remote Sensing, Georgia*
- *Prof. Byung-Gyu Kim (Congress Steering Committee); Multimedia Processing Communications Lab.(MPCL), Department of Computer Science and Engineering, College of Engineering, SunMoon University, South Korea*
- *Prof. Tai-hoon Kim; School of Information and Computing Science, University of Tasmania, Australia*
- *Prof. Louie Lolong Lacatan; Chairperson, Computer Engineering Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria*
- *Prof. Dr. Guoming Lai; Computer Science and Technology, Sun Yat-Sen University, Guangzhou, P. R. China*
- *Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria*
- *Prof. James J. (Jong Hyuk) Park (Congress Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Dr. Akash Singh (Congress Steering Committee); IBM Corporation, Sacramento, California, USA; Chartered Scientist, Science Council, UK; Fellow, British Computer Society; Member, Senior IEEE, AACR, AAAS, and AAAI; IBM Corporation, USA*

- *Chiranjibi Sitaula; Head, Department of Computer Science and IT, Ambition College, Kathmandu, Nepal*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Prof. Dr. Ir. Sim Kok Swee; Fellow, IEM; Senior Member, IEEE; Faculty of Engineering and Technology, Multimedia University, Melaka, Malaysia*
- *Prof. Fernando G. Tinetti (Congress Steering Committee); School of Computer Science, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Prof. Quoc-Nam Tran (Congress Steering Committee); Department of Computer Science, Southeastern Louisiana University, Louisiana, USA*
- *Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine*
- *Prof. Shiu-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications*
- *Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Foundations of Computer Science

FCS 2020 – Program Committee

- *Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Prof. Juan Jose Martinez Castillo; Director, The Acanatelys Alan Turing Nikola Tesla Research Group and GIPEB, Universidad Nacional Abierta, Venezuela*
- *Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science & Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Zhangisina Gulnur Davletzhanovna; Vice-rector of the Science, Central-Asian University, Kazakhstan, Almaty, Republic of Kazakhstan; Vice President of International Academy of Informatization, Kazakhstan, Almaty, Republic of Kazakhstan*
- *Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Prof. Tai-hoon Kim; School of Information and Computing Science, University of Tasmania, Australia*
- *Prof. Dr. Guoming Lai; Computer Science and Technology, Sun Yat-Sen University, Guangzhou, P. R. China*
- *Dr. Vitus S. W. Lam; Senior IT Manager, Information Technology Services, The University of Hong Kong, Kennedy Town, Hong Kong; Chartered Member of The British Computer Society, UK; Former Vice Chairman of the British Computer Society (Hong Kong Section); Chartered Engineer & Fellow of the Institution of Analysts and Programmers*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Tech., Ambrose Alli University, Edo State, Nigeria*

- *Chiranjibi Sitaula; Head, Department of Computer Science and IT, Ambition College, Kathmandu, Nepal*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Dr. Tse Guan Tan; Faculty of Creative Technology and Heritage, Universiti Malaysia Kelantan, Kelantan, Malaysia*
- *Prof. Fernando G. Tinetti (Congress Steering Committee); School of Computer Science, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Varun Vohra; Certified Information Security Manager (CISM); Certified Information Systems Auditor (CISA); Associate Director (IT Audit), Merck, New Jersey, USA*
- *Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

Software Engineering Research and Practice

SERP 2020 – Program Committee

- *Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); Electrical & Computer Engineering Department; Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Michigan, USA*
- *Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Dr. Travis Atkison; Director, Digital Forensics and Control Systems Security Lab, Department of Computer Science, College of Engineering, The University of Alabama, Tuscaloosa, Alabama, USA*
- *Dr. Azita Bahrami (Vice-Chair); President, IT Consult, USA*
- *Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de València (UPV), Department of Computer Engineering (DISCA), Valencia, Spain*
- *Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Zhangisina Gulnur Davletzhanovna; Vice-rector of the Science, Central-Asian University, Kazakhstan, Almaty, Republic of Kazakhstan; Vice President of International Academy of Informatization, Kazakhstan, Almaty, Republic of Kazakhstan*
- *Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Dr. Lamia Atma Djoudi (Chair, Doctoral Colloquium & Demos Sessions); France*
- *Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California,*

USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA

- *Prof. Ray Hashemi (Vice-Chair); College of Engineering and Computing, Georgia Southern University, Georgia, USA*
- *Prof. Byung-Gyu Kim (Congress Steering Committee); Multimedia Processing Communications Lab.(MPCL), Department of Computer Science and Engineering, College of Engineering, SunMoon University, South Korea*
- *Prof. Louie Lolong Lacatan; Chairperson, CE Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, International Association of Online Engineering (IAOE), Austria*
- *Dr. Vitus S. W. Lam; Senior IT Manager, Information Technology Services, The University of Hong Kong, Kennedy Town, Hong Kong; Chartered Member of The British Computer Society, UK; Former Vice Chairman of the British Computer Society (Hong Kong Section); Chartered Engineer & Fellow of the Institution of Analysts and Programmers*
- *Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations*
- *Prof. Dr., Eng. Robert Ehimen Okonigene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Technology, Ambrose Alli University, Nigeria*
- *Prof. James J. (Jong Hyuk) Park (Congress Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Prof. Dr. R. Ponalagusamy; Department of Mathematics, National Institute of Technology, India*
- *Prof. Abd-El-Kader Sahraoui; Toulouse University and LAAS CNRS, Toulouse, France*
- *Dr. Akash Singh (Congress Steering Committee); IBM Corporation, Sacramento, California, USA; Chartered Scientist, Science Council, UK; Fellow, British Computer Society; Member, Senior IEEE, AACR, AAAS, and AAAI; IBM Corporation, USA*
- *Chiranjibi Sitaula; Head, Department of Computer Science and IT, Ambition College, Kathmandu, Nepal*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Prof. Fernando G. Tinetti (Congress Steering Committee); School of CS, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department,*

Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine

- *Varun Vohra; Certified Information Security Manager (CISM); Certified Information Systems Auditor (CISA); Associate Director (IT Audit), Merck, New Jersey, USA*
- *Dr. Haoxiang Harry Wang (CSCE); Cornell University, Ithaca, New York, USA; Founder and Director, GoPerception Laboratory, New York, USA*
- *Prof. Shiuh-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.*
- *Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*
- *Prof. Jane You (Congress Steering Committee); Associate Head, Department of Computing, The Hong Kong Polytechnic University, Kowloon, Hong Kong*

e-Learning, e-Business, Enterprise Information Systems, & e-Government

EEE 2020 – Program Committee

- *Prof. Abbas M. Al-Bakry (Congress Steering Committee); University President, University of IT and Communications, Baghdad, Iraq*
- *Prof. Emeritus Nizar Al-Holou (Congress Steering Committee); ECE Department; Vice Chair, IEEE/SEM-Computer Chapter; University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Emeritus Hamid R. Arabnia (Congress Steering Committee); The University of Georgia, USA; Editor-in-Chief, Journal of Supercomputing (Springer); Fellow, Center of Excellence in Terrorism, Resilience, Intelligence & Organized Crime Research (CENTRIC).*
- *Dr. Azita Bahrami (Vice-Chair); President, IT Consult, USA*
- *Prof. Dr. Juan-Vicente Capella-Hernandez; Universitat Politècnica de València (UPV), Department of Computer Engineering (DISCA), Valencia, Spain*
- *Prof. Emeritus Kevin Daimi (Congress Steering Committee); Department of Mathematics, Computer Science and Software Engineering, University of Detroit Mercy, Detroit, Michigan, USA*
- *Prof. Zhangisina Gulnur Davletzhanovna; Vice-rector of the Science, Central-Asian University, Kazakhstan, Almaty, Republic of Kazakhstan; Vice President of International Academy of Informatization, Kazakhstan, Almaty, Republic of Kazakhstan*
- *Prof. Leonidas Deligiannidis (Congress Steering Committee); Department of Computer Information Systems, Wentworth Institute of Technology, Boston, Massachusetts, USA*
- *Prof. Mary Mehrnoosh Eshaghian-Wilner (Congress Steering Committee); Professor of Engineering Practice, University of Southern California, California, USA; Adjunct Professor, Electrical Engineering, University of California Los Angeles, Los Angeles (UCLA), California, USA*
- *Prof. George A. Gravvanis (Congress Steering Committee); Director, Physics Laboratory & Head of Advanced Scientific Computing, Applied Math & Appli-*

cations Research Group; Professor of Applied Mathematics and Numerical Computing and Department of ECE, School of Engineering, Democritus University of Thrace, Xanthi, Greece.

- *Prof. Houcine Hassan; Department of Computer Engineering (Systems Data Processing and Computers), Universitat Politecnica de Valencia, Spain*
- *Prof. George Jandieri (Congress Steering Committee); Georgian Technical University, Tbilisi, Georgia; Chief Scientist, The Institute of Cybernetics, Georgian Academy of Science, Georgia; Ed. Member, International Journal of Microwaves and Optical Technology, The Open Atmospheric Science Journal, American Journal of Remote Sensing, Georgia*
- *Prof. Dr. Abdeldjalil Khelassi; CS Department, Abou beker Belkaid University of Tlemcen, Algeria; Editor-in-Chief, Medical Tech. Journal; Assoc. Editor, Electronic Physician Journal (EPJ) - Pub Med Central*
- *Prof. Byung-Gyu Kim (Congress Steering Committee); Multimedia Processing Communications Lab.(MPCL), Department of CSE, College of Engineering, SunMoon University, South Korea*
- *Prof. Louie Lolong Lacatan; Chairperson, Computer Engineering Department, College of Engineering, Adamson University, Manila, Philippines; Senior Member, International Association of Computer Science and Information Technology (IACSIT), Singapore; Member, IAOE, Austria*
- *Dr. Andrew Marsh (Congress Steering Committee); CEO, HoIP Telecom Ltd (Healthcare over Internet Protocol), UK; Secretary General of World Academy of BioMedical Sciences and Technologies (WABT) a UNESCO NGO, The United Nations*
- *Dr. Ali Mostafaeipour; Industrial Engineering Department, Yazd University, Yazd, Iran*
- *Dr. Housseem Eddine Nouri; Informatics Applied in Management, Institut Supérieur de Gestion de Tunis, University of Tunis, Tunisia*
- *Prof. Dr., Eng. Robert Ehimen Okongene (Congress Steering Committee); Department of Electrical & Electronics Engineering, Faculty of Engineering and Tech., Ambrose Alli University, Edo State, Nigeria*
- *Prof. James J. (Jong Hyuk) Park (Congress Steering Committee); Department of Computer Science and Engineering (DCSE), SeoulTech, Korea; President, FTRA, EiC, HCIS Springer, JoC, IJITCC; Head of DCSE, SeoulTech, Korea*
- *Ashu M. G. Solo (Publicity), Fellow of British Computer Society, Principal/R&D Engineer, Maverick Technologies America Inc.*
- *Dr. Vivian Saltan, California State University, Los Angeles, California, USA*
- *Prof. Fernando G. Tinetti (Congress Steering Committee); School of Computer Science, Universidad Nacional de La Plata, La Plata, Argentina; also at Comision Investigaciones Cientificas de la Prov. de Bs. As., Argentina*
- *Prof. Hahanov Vladimir (Congress Steering Committee); Vice Rector, and Dean of the Computer Engineering Faculty, Kharkov National University of Radio Electronics, Ukraine and Professor of Design Automation Department, Computer Engineering Faculty, Kharkov; IEEE Computer Society Golden Core Member; National University of Radio Electronics, Ukraine*

- *Prof. Shiuh-Jeng Wang (Congress Steering Committee); Director of Information Cryptology and Construction Laboratory (ICCL) and Director of Chinese Cryptology and Information Security Association (CCISA); Department of Information Management, Central Police University, Taoyuan, Taiwan; Guest Ed., IEEE Journal on Selected Areas in Communications.*
- *Prof. Layne T. Watson (Congress Steering Committee); Fellow of IEEE; Fellow of The National Institute of Aerospace; Professor of Computer Science, Mathematics, and Aerospace and Ocean Engineering, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA*

Contents

Part I Curriculum Design, Academic Content, and Learning Objectives

Empirical Analysis of Strategies Employed Within an ICT Curriculum to Increase the Quantity of Graduates	3
Nicole Herbert, Erik Wapstra, David Herbert, Kristy de Salas, and Tina Acuña	
Incorporating Computer Programming into Mathematics Curricula to Enhance Learning for Low-Performing, Underserved Students	17
Alan Shaw and William Crombie	
Examining the Influence of Participating in a Cyber Defense Track on Students' Cybersecurity Knowledge, Awareness, and Career Choices	31
Michelle Peters, T. Andrew Yang, Wei Wei, Kewei Sha, and Sadegh Davari	
Team-Based Online Multidisciplinary Education on Big Data + High-Performance Computing + Atmospheric Sciences	43
Jianwu Wang, Matthias K. Gobbert, Zhibo Zhang, and Aryya Gangopadhyay	
Integrating the Development of Professional Skills Throughout an ICT Curriculum Improves a Graduate's Competency	55
Nicole Herbert, David Herbert, Erik Wapstra, Kristy de Salas, and Tina Acuña	
Preparing Computing Graduates for the Workplace: An Assessment of Relevance of Curricula to Industry	69
Ioana Chan Mow, Elisapeta Mauai, Vaisualua Okesene, and Ioana Sinclair	

Benchmarking the Software Engineering Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE Software Engineering Body of Knowledge (SWE Knowledge Areas #6 –10)..... 85
 Moh'd A. Radaideh

Part II Educational Tools, Novel Teaching Methods and Learning Strategies

Design for Empathy and Accessibility: A Technology Solution for Deaf Curling Athletes..... 103
 Marcia R. Friesen, Ryan Dion, and Robert D. McLeod

An Investigation on the Use of WhatsApp Groups as a Mobile Learning System to Improve Undergraduate Performance 117
 A. Rushane Jones and B. Sherrene Bogle

Using Dear Data Project to Introduce Data Literacy and Information Literacy to Undergraduates 131
 Vetricia L. Byrd

An Educational Tool for Exploring the Pumping Lemma Property for Regular Languages 143
 Josue N. Rivera and Haiping Xu

An Educational Guide to Creating Your Own Cryptocurrency 163
 Paul Medeiros and Leonidas Deligiannidis

Peer Assistant Role Models in a Graduate Computer Science Course..... 179
 Evava Pietri, Leslie Ashburn-Nardo, and Snehasis Mukhopadhyay

A Project-Based Approach to Teaching IoT 195
 Varick L. Erickson, Pragya Varshney, and Levent Ertaul

Computational Thinking and Flipped Classroom Model for Upper-Division Computer Science Majors 217
 Antonio-Angel L. Medel, Anthony C. Bianchi, and Alberto C. Cruz

A Dynamic Teaching Learning Methodology Enabling Fresh Graduates Starting Career at Mid-level 229
 Abubokor Hanip and Mohammad Shahadat Hossain

Innovative Methods of Teaching the Basic Control Course 249
 L. Keviczky, T. Vámos, A. Benedek, R. Bars, J. Hetthéssy, Cs. Bányász, and D. Sik

Part III Frontiers in Education – Methodologies, Student Academic Preparation and Related Findings

Towards Equitable Hiring Practices for Engineering Education Institutions: An Individual-Based Simulation Model 265
 Marcia R. Friesen and Robert D. McLeod

Developing a Scalable Platform and Analytics Dashboard for Manual Physical Therapy Practices Using Pressure Sensing Fabric 277
 Tyler V. Rimaldi, Daniel R. Grossmann, and Donald R. Schwartz

Tracking Changing Perceptions of Students Through a Cyber Ethics Course on Artificial Intelligence 287
 Zeenath Reza Khan, Swathi Venugopal, and Farhad Oroumchian

Predicting the Academic Performance of Undergraduate Computer Science Students Using Data Mining 303
 Faiza Khan, Gary M. Weiss, and Daniel D. Leeds

An Algorithm for Determining if a BST Node’s Value Can Be Changed in Place 319
 Daniel S. Spiegel

Class Time of Day: Impact on Academic Performance 327
 Suzanne C. Wagner, Sheryl J. Garippo, and Petter Lovaaas

A Framework for Computerization of Punjab Technical Education System for Financial Assistance to Underrepresented Students 337
 Harinder Pal Singh and Harpreet Singh

Parent-Teacher Portal (PTP): A Communication Tool 351
 Mudasser F. Wyne, Matthew Hunter, Joshua Moran, and Babita Patil

Part IV Foundations of Computer Science: Architectures, Algorithms, and Frameworks

Exact Floating Point 365
 Alan A. Jorgensen and Andrew C. Masters

Random Self-modifiable Computation 375
 Michael Stephen Fiske

ECM Factorization with QRT Maps 395
 Andrew N.W. Hone

What Have Google’s Random Quantum Circuit Simulation Experiments Demonstrated About Quantum Supremacy? 411
 Jack K. Horner and John F. Symons

Chess Is Primitive Recursive 421
 Vladimir A. Kulyukin

How to Extend Single-Processor Approach to Explicitly Many-Processor Approach 435
 János Vég

Formal Specification and Verification of Timing Behavior in Safety-Critical IoT Systems 459
 Yangli Jia, Zhenling Zhang, Xinyu Cao, and Haitao Wang

Introducing Temporal Behavior to Computing Science 471
 János Vég

Evaluation of Classical Data Structures in the Java Collections Framework 493
 Anil L. Pereira

Part V Software Engineering, Dependability, Optimization, Testing, and Requirement Engineering

Securing a Dependability Improvement Mechanism for Cyber-Physical Systems 511
 Gilbert Regan, Fergal Mc Caffery, Pangkaj Chandra Paul, Ioannis Sorokos, Jan Reich, Eric Armengaud, and Marc Zeller

A Preliminary Study of Transactive Memory System and Shared Temporal Cognition in the Collaborative Software Process Tailoring 523
 Pei-Chi Chen, Jung-Chieh Lee, and Chung-Yang Chen

Mixed-Integer Linear Programming Model for the Simultaneous Unloading and Loading Processes in a Maritime Port 533
 Ali Skaf, Sid Lamrous, Zakaria Hammoudan, and Marie-Ange Manier

How to Test Interoperability of Different Implementations of a Complex Military Standard 545
 Andre Schöbel, Philipp Klotz, Christian Zschke, and Barbara Essendorfer

Overall Scheduling Requirements for Scheduling Synthesis in Automotive Cooperative Development 557
 Arthur Strasser, Christoph Knieke, and Andreas Rausch

Extracting Supplementary Requirements for Energy Flexibility Marketplace 567
 Tommi Aihkisalo, Kristiina Valtanen, and Klaus Känsälä

A Dynamic Scaling Methodology for Improving Performance of Data-Intensive Systems 577
 Nashmiah Alhamdawi and Yi Liu

Part VI Software Engineering Research, Practice, and Novel Applications

Technical Personality as Related to Intrinsic Personality Traits 597
 Marwan Shaban, Craig Tidwell, Janell Robinson, and Adam J. Rocke

Melody-Based Pitch Correction Model for a Voice-Driven Musical Instrument 609
 John Carelli

Analysis of Bug Types of Textbook Code with Open-Source Software 629
 Young Lee and Jeong Yang

Implications of Blockchain Technology in the Health Domain..... 641
 Merve Vildan Baysal, Özden Özcan-Top, and Aysu Betin Can

A Framework for Developing Custom Live Streaming Multimedia Apps . 657
 Abdul-Rahman Mawlood-Yunis

Change Request Prediction in an Evolving Legacy System: A Comparison 671
 Lamees Alhazzaa and Anneliese Amschler Andrews

Using Clients to Support Extract Class Refactoring 695
 MUSAAD ALZAHrani

Analyzing Technical Debt of a CRM Application by Categorizing Ambiguous Issue Statements 705
 Yasemin Doğancı, Özden Özcan-Top, and Altan Koçyiğit

Applying DevOps for Distributed Agile Development: A Case Study 719
 Asif Qumer Gill and Devesh Maheshwari

Water Market for Jazan, Saudi Arabia 729
 Fathe Jeribi, Sungchul Hong, and Ali Tahir

Modeling Unmanned Aircraft System Maintenance Using Agile Model-Based Systems Engineering..... 741
 Justin R. Miller, Ryan D. L. Engle, Brent T. Langhals, Michael R. Grimaila, and Douglas D. Hodson

Benchmarking the Software Engineering Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE Software Engineering Body of Knowledge (Software Engineering Knowledge Areas #1 –5)..... 747
 Moh’d A. Radaideh

A Study of Third-Party Software Compliance and the Associated Cybersecurity Risks..... 769
 Rashel Dibi, Brandon Gilchrist, Kristen Hodge, Annicia Woods, Samuel Olatunbosun, and Taiwo Ajani

Further Examination of YouTube’s Rabbit-Hole Algorithm 775
 Matthew Moldawsky

Part VII Educational Frameworks and Strategies, and e-Learning

Characterizing Learner’s Comments and Rating Behavior in Online Course Platforms at Scale 781
 Mirko Marras and Gianni Fenu

Supporting Qualification Based Didactical Structural Templates for Multiple Learning Platforms 793
 Michael Winterhagen, Minh Duc Hoang, Benjamin Wallenborn, Dominic Heutelbeck, and Matthias L. Hemmje

Enhancing Music Teachers’ Cognition and Metacognition: Grassroots FD Project 2019 at Music College 809
 Chiharu Nakanishi, Asako Motojima, and Chiaki Sawada

Scalable Undergraduate Cybersecurity Curriculum Through Auto-graded E-Learning Labs 825
 Aspen Olmsted

The Effect of Matching Learning Material to Learners’ Dyslexia Type on Reading Performance 837
 Hadeel Al-Dawsari and Robert Hendley

Individualized Educational System Supporting Object-Oriented Programming 847
 F. Fischman, H. Lersch, M. Winterhagen, B. Wallenborn, M. Fuchs, M. Then, and M. Hemmje

Part VIII e-Business, Enterprise Information Systems, and e-Government

Emerging Interactions of ERP Systems, Big Data and Automotive Industry 863
 Florie Bandara and Uchitha Jayawickrama

Software Evaluation Methods to Support B2B Procurement Decisions: An Empirical Study 879
 F. Bodendorf, M. Lutz, and J. Franke

Sentiment Analysis of Product Reviews on Social Media 899
 Velam Thanu and David Yoon

Research on Efficient and Fuzzy Matching Algorithm in Information Dissemination System 909
 Qinwen Zuo, Fred Wu, Fei Yan, Shaofei Lu, Colmenares-diaz Eduardo, and Junbin Liang

Agile IT Service Management Frameworks and Standards: A Review 921
M. Mora, J. Marx-Gomez, F. Wang, and O. Diaz

Contingency Planning: Prioritizing Your Resources 937
Kathryne Burton, Necole Cuffee, Darius Neclos, Samuel Olatunbosun,
and Taiwo Ajani

Smart Low-Speed Self-Driving Transportation System 943
Zhenghong Wang and Bowu Zhang

Are Collaboration Tools Safe? An Assessment of Their Use and Risks 949
Cquoya Haughton, Maria Isabel Herrmann, Tammi Summers,
Sonya Worrell, Samuel Olatunbosun, and Taiwo Ajani

Tourism Service Auction Market for Saudi Arabia 961
Saad Nasser Almutwa and Sungchul Hong

The Use of Crowdsourcing as a Business Strategy 971
Hodaka Nakanishi and Yuko Syozugawa

Index 985

Part I
Curriculum Design, Academic Content,
and Learning Objectives

Empirical Analysis of Strategies Employed Within an ICT Curriculum to Increase the Quantity of Graduates



Nicole Herbert, Erik Wapstra, David Herbert, Kristy de Salas, and Tina Acuña

1 Introduction

The University of Tasmania (UTAS) commenced a curriculum renewal process in 2012. At the time, there was concern both within the information and communication technology (ICT) industry and within government agencies about the number of the ICT graduates [1, 2]. Potential students had incorrect perceptions of the field of ICT [3], and this resulted in low commencement rates for ICT higher education courses in comparison to other disciplines [4]. High attrition rates in ICT courses, caused by a number of factors mostly relating to a lack of student engagement [5], motivation [6], and academic success [7, 8] were also impacting on the number of graduates.

This chapter reports on a broad and deep ICT curriculum change and uses data collected over a 9-year time period to conduct an empirical evaluation of the changes to the quantity of graduates. This chapter contributes to the field of ICT curriculum design as it provides implementation techniques for strategies that can have positive long-term outcomes. The research question explored is: *What is the impact of strategies designed to amend misconceptions and improve perceptions, motivation, engagement, and academic success on the quantity of graduates?*

N. Herbert (✉) · E. Wapstra · D. Herbert · K. de Salas · T. Acuña
University of Tasmania, Hobart, TAS, Australia
e-mail: Nicole.Herbert@utas.edu.au

© Springer Nature Switzerland AG 2021
H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_1

2 Related Work

Rapidly evolving technology has resulted in a continuous demand for competent ICT graduates. In 2019, the Australian Computer Society (ACS) released figures forecasting that Australia will require an additional 100,000 ICT specialist workers by 2024 [9]. A potential source of these workers is tertiary ICT graduates.

While the ACS reported that domestic undergraduate enrolments rose from a low of around 19,000 in 2010 to 30,000 in 2017 [9], this growth is not large enough to meet the forecasted demand, and it is further degraded by a high attrition rate. Even though there has been steady growth in completions since 2012, there were only 4400 domestic undergraduate completions in 2017 [9]. It is imperative that domestic completion rates in ICT courses improve for the growth of the ICT sector.

International students that graduate from ICT courses in Australia are also a potential source of skilled employees. In 2012, the growth in international student commencements in ICT courses had stagnated and started to decline [4], though since then, there has been significant growth, with international students comprising 39% of the national ICT undergraduate student population [9]. Similar to domestic graduates, there has been steady growth in completions since 2012, with 4000 international undergraduate completions in 2017 [9]. Even with this growth in total graduates, the supply of ICT employees from domestic and international graduates is much smaller than the predicted increase in the size of Australia's technology workforce over the next decade [9].

It is well recognized that students choose not to study ICT due to their perceptions of the field [3, 10–13]. A career in ICT is perceived as male-dominated, repetitive, isolated, and focused on the technical rather than the professional [3]. While this perception was valid in the past, the industry has transformed, and potential applicants need to be aware of how fulfilling an ICT career can be and how diverse the opportunities are.

To increase the quantity of ICT graduates, it is necessary to not only increase the commencements in ICT courses but also reduce the rate of course attrition. National attrition across all disciplines was around 17% in 2012 [4], in comparison to a national attrition rate of 43% for ICT courses [1]. There have been a number of studies identifying the causes of this high course attrition.

Poor course choice due to student misconceptions of what ICT is and what is involved in studying ICT is a leading cause of course attrition [3, 10–13]. Beaubouef et al. [11] summarized a number of misconceptions that can impact on both course commencements and attrition:

Nature of the field – ICT is much wider than producing reports and collating data and infiltrates a wide range of industries.

ICT is easy – ICT requires maths and problem-solving skills and a disciplined approach to solve complicated problems.

Social issues and communication skills – ICT careers are not solitary positions and require written and oral communication skills to convey ideas and concepts to develop systems that meet user requirements.

Programming – while it is essential that all ICT personnel have some ability to program, it is only one of the many important skills required. Biggers et al. [12] found that, although the primary reason students gave for leaving was allegedly a loss of interest, the underlying explanation was often related to the undesirability of a programming-only career.

One difference between students who complete and students who leave is their motivation to study [6, 14]. Providing evidence that the course can result in a secure, satisfying, and financially rewarding career can influence the decision to continue [14, 15]. Smith et al. [8] found that a lack of academic success is also a major factor in the decision to withdraw and ICT students who pass subjects were more likely to continue. A lack of student engagement in ICT courses was also found to be a leading cause of course attrition, particularly for first-year students [5, 16]. This was often a result of poor course design: poor quality teaching, feedback, or course structure [7, 10, 13, 14, 16–19], poorly related practical work to professional practice [10, 13, 14, 17–19], and low levels of interaction with peers and staff [6, 12–14, 18, 20].

3 The UTAS Situation

The University of Tasmania (UTAS) is responsible for developing competent graduates for a broad local ICT industry. In 2014, the Tasmanian ICT sector employed over 4500 people and generated industry value add of around \$640 million, representing less than 1.6% of the Australian ICT sector’s total [21]. The Tasmanian ICT sector had been constrained by skills shortages for a decade [21].

As had been the case for ICT courses nationally, the student numbers had stagnated, and the attrition rates were high, as shown in Table 1. As Tasmania is an island state with a small population, there is a very limited domestic market of tertiary applicants. There was reliance on international student enrolments, but these were in decline – down to 22% in 2012 [4]. 11% of the students were enrolled in the Bachelor of Information Systems (BIS), while the rest were in the Bachelor of Computing (BComp). The attrition rate in the UTAS ICT courses prior to 2013 was 57%, much higher than the national ICT course average of 43% in 2012 [1].

Table 1 Student data for the BComp/BIS

	2010	2011	2012	2013
Commencing students	131	137	135	167
Domestic student ratio	73%	69%	78%	78%
Attrition rate	55%	54%	63%	74%

4 The Case Study Curriculum

It was a recommendation of a course review panel that the current two undergraduate courses be discontinued and a single course be created in the belief a new course would attract and retain more students. The curriculum renewal design process commenced in 2012, and the resultant curriculum, the Bachelor of Information and Communication Technology (BICT), was first offered in 2014 [22]. The curriculum was aligned with the Association for Computing Machinery (ACM) information technology (IT) curriculum, but included aspects from the ACM information systems (IS) and ACM computer science (CS) curricula [23] to encompass study across ICT disciplines to ensure coverage of complementary knowledge.

Discussions with more than 30 local ICT industry members revealed that they were satisfied with the ICT graduate's technical skills but concerned by the weaker professional skills, such as communication and collaboration [22]. This led to the design of an ICT curriculum that integrated the development of professional skills alongside the development of the technical skills (such as programming, networking, security, and databases) and nontechnical ICT skills (project management and business analysis) to create graduates with a strong employability skill set for an identified range of career outcomes [22]. It was essential that as well as increasing the number of graduates that the technical skill competency levels were maintained and professional skill competency improved. There was no desire to improve graduation rates by lowering standards. A concurrent longitudinal study analyzed cohorts of students from 2012 to 2018 and concluded that this case study curriculum had improved the students' competency with professional skills without having a detrimental impact on their competency with technical skills [24].

This case study ICT curriculum also employed strategies to increase the number of ICT graduates. These strategies were based on recommendations from prior research which were often untested or evaluated over only a few subjects or short trial periods. Strategies were employed curriculum-wide to amend student misconceptions of ICT, improve student perceptions of ICT, and improve student motivation, engagement, and academic success. This study contributes to the field of ICT curriculum design as it provides implementation techniques for the strategies and conducts an empirical evaluation over an extended time period to ascertain the changes to the quantity of graduates.

4.1 Amending Misconceptions and Improving Perceptions and Motivation

The main strategy to increase course commencements for the case study curriculum centered on using the transformation of graduate ICT career outcomes that resulted from the rapidly changing nature of the ICT industry to improve student perceptions of ICT careers. There were concerns that potential students lacked awareness of the

wide range of career possibilities in ICT [12]. There were also concerns for the low enrolments in years 11 and 12 in ICT-related subjects [2] and the flow on effect this was having at the tertiary level. To increase commencements, there was a need for ICT higher education courses to appeal to students who did and who did not have prior learning in ICT.

To identify a set of relevant graduate career outcomes, discussions were held with local industry members [25]. To facilitate multiple career outcomes, the Skills Framework for the Information Age (SFIA) was used throughout the design process to identify the technical and nontechnical skills for the graduate employability skill set [25]. The skill development was then integrated throughout the curriculum [22]. The careers included not just programmers but modelers, developers, designers, analysts, administrators, and managers; 30 different ICT career outcomes were identified to improve perceptions [25]. As well as attracting a broader range of applicants, it was hoped this approach would also assist in retaining students who found programming to be challenging or unengaging by correcting misconceptions and motivating them toward another ICT career.

To attract more students, a course structure was chosen that ensured all graduates received a compulsory core that developed skills for a broad range of careers but gave students flexibility to choose a pathway that enabled them to develop deeper skills for their chosen ICT career outcome [22]. The renewed curriculum included a compulsory ICT Professional major (eight units of integrated study across 3 years – a unit is a subject) structured to amend misconceptions by including foundational technical knowledge (in the areas of programming and mathematics) with nontechnical business skills and culminating in an authentic capstone experience where students designed and developed software for industry clients. The ICT Professional major sat alongside the IT minor (four units of integrated study across 2 years) that provided technical skills in databases, web systems, networks, and operating systems with cybersecurity integrated within each unit. To motivate students, they chose their second major from either a Games and Creative Technology (GCT) major or a Software Development (SD) major. Both majors provided depth in programming and system design, though the GCT major used programming languages and development processes suitable for game-related career outcomes such as games developer/designer. The SD major was broader, aiming for career outcomes ranging from software designer/developer to system administrator to business analyst. The course structure also included 4 elective units, for a total of 24 units.

To improve the student perceptions about ICT, a module was included in a first-year unit to motivate the students to complete their course by exploring the available careers [5, 6, 15], demonstrating that an ICT career was more than coding [12]. The students identified when and where their knowledge and skills for each career outcome would be developed within the course [26].

To attract more students and to correct misconceptions and improve student perceptions and motivation, the course was given a workplace and professional focus [15, 18]. To strengthen the student's sense of relevance about their course, a balance between the technical and nontechnical focus was created, alongside a

balance between practical application and theory [18, 22]. Each semester had at least one unit that had a nontechnical or professional focus, to give the curriculum a workplace or business focus, allowing students to make the connections with professional practice [24].

4.2 Improving Engagement

It has been identified by many researchers that a lack of student engagement leads to course attrition [5, 13, 16]. To improve student engagement with the renewed curriculum, new teaching practices and technology-enhanced delivery were incorporated throughout the course, allowing a student-centered learning experience which has been shown to reduce course attrition [16]. All units involved online materials to allow self-paced learning. While a few units went completely online, most units retained practical hands-on tutorials of at least 2 hours per week to enable active learning with tutor support. Tutorial exercises and assignments were related to professional practice using real-world scenarios [13, 18, 19]. Most units reduced face-to-face lectures to 1 to 2 hours per week, and these were also recorded and included in the online materials. Some removed lectures entirely to adopt a flipped classroom delivery style where students learn by applying the theoretical content, obtained via online materials prior to the tutorial, in a variety of cooperative activities in small classes [26].

Social integration and establishing relationships and having student-student interactions have been shown to impact student engagement and are an influencing factor in the decision to persist toward completion [6, 12–14, 18, 20]. Group work was introduced throughout the curriculum to foster the development of relationships and community at all levels [12]; at least one unit each semester involved significant groupwork [24]. A core unit undertaken by most students in their first semester used a flipped-classroom style of delivery and focused on group work, with the students placed in many random groups, but also some they formed themselves [26]. Students established supportive peer groups in their first semester, which increased confidence and reduced feelings of isolation to encourage them to continue [14].

4.3 Improving Academic Success

A number of prior studies have shown that academic success is a major factor in the decision to withdraw, particularly in the first year [7, 8, 27]. To allow students to achieve some academic success and identify areas of ICT in which they have aptitude to motivate them toward an ICT career, the course structure was arranged so that the students would engage with a broad range of topics each semester [12]. Rather than four units focused on developing programming skills, students would have one programming unit, one IT-related technical unit,

one professional/business-focused unit, and one elective each semester in the first 2 years. In particular, students undertook a range of ICT-related topics in the first year, to increase not only the level of academic success and motivation to complete but also the level of engagement. In addition to programming, operating systems, and databases, five new units were created for the first year, covering a range of relevant topics: mathematics, emerging technology, ICT professionalism, games, and artificial intelligence.

The correlation between prior programming experience and the level of academic success in introductory programming has been widely studied [20]. Research has shown that when both novice and experienced programmers are enrolled in the same introductory programming subject, the novice programmers are negatively impacted due to a lack of confidence [12], and a lack of confidence is a core factor in the decision to withdraw [14, 27]. In the previous curriculum, all students commenced with a foundational programming unit, and if they passed, they advanced to a second introductory programming unit. This resulted in some highly skilled students performing poorly or withdrawing due to lack of engagement, as they had already acquired this foundational level programming knowledge. Within the new curriculum, the students that were entering the course with strong prior programming knowledge (evaluated by their grade in pre-tertiary programming subjects), were allowed to replace the foundational programming unit by an ICT elective and advance directly to the second introductory programming unit. This allowed the students that did not have this prior experience the opportunity to develop foundational programming skills. A strong foundation leads to higher marks, thus increasing their satisfaction with the course [18, 27], and their confidence was not eroded as they were only working with peers of similar ability [12].

During the early years of the renewed curriculum, the foundational programming skills unit had a high failure rate. In 2017, a change was made to the teaching and assessment style in this unit to help students achieve the learning outcomes. Rather than a significantly weighted end-of-unit exam, continuous assessment was introduced. Students were required to complete programming tasks on an almost weekly basis, with regular formative feedback (re-submission was allowed) to improve the development of their skills. The students were also required to pass two supervised individual short tests held toward the middle and end of the semester. The learning outcomes remained consistent; students still had to achieve the same level of competency with programming over the same period of time [24].

5 Method

It is difficult to perform a controlled study, even within one institution, over such a substantial period of time. During this time, the School at UTAS responsible for the ICT courses has undergone many personnel, management, and structural changes, the institution has had three Vice Chancellors, and there have been many government higher education policy and practice changes.

Controlling as much as possible, this study evaluates the overall change on the quantity of ICT graduates over an extended period; it examines both domestic and international student populations.

A chi-squared test was used on the categorical data to test for differences between implementation periods of the previous and renewed curriculum. This study used data from 2010 to 2013, the final years of the previous curriculum, and compared it to data from 2014 to 2018, the years of the renewed curriculum. The student data is summarized in Table 1 (BComp/BIS) and Table 2 (BICT).

6 Results

Figure 1 illustrates the ICT student commencements as well as the rate of course attrition over the timeframe of the study.

While Fig. 1 makes it appear that commencement rates improved in 2014, this is due to double counting caused by 52 students who transferred from the previous ICT courses to the BICT in 2014. This transfer was encouraged, so that students could experience the new content for their remaining years of study. There were also another 22 students who transferred from the previous courses from 2015 to 2018. This means that from 2010 to 2013, 570 new students commenced a BComp/BIS course, and from 2014 to 2017, only 562 new students commenced the BICT course.

Table 2 Student data for the BICT

	2014	2015	2016	2017	2018
Commencing students	186	178	115	155	170
Domestic student ratio	74%	56%	70%	54%	44%
Attrition rate	61%	61%	45%	26%	22%

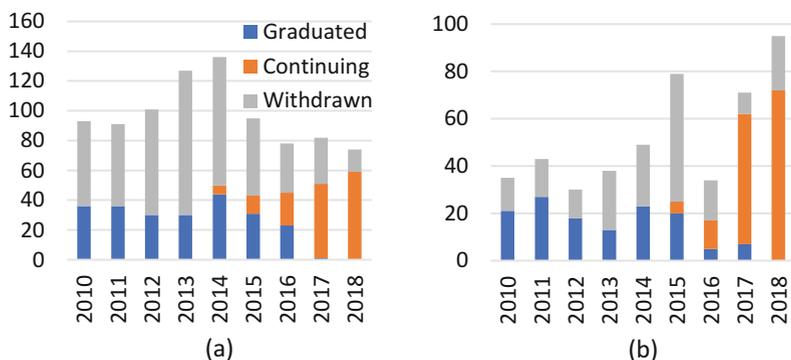


Fig. 1 Yearly student outcomes: (a) Domestic students. (b) International students

There was a significant difference in the course attrition rates between 2010 and 2012 when it was 57% and 2017 and 2018 when it was 24% ($\chi^2(1) = 80.193$, p -value <0.00001). Between 2017 and 2018, students have not had as much time to withdraw.

To evaluate changes in academic success, the overall pass rate for the first year was examined. There was a significant difference in the number of students that passed all their first-year units from 2010 to 2012 when 33% of students passed all units and from 2017 to 2018 when 49% of students passed all units ($\chi^2(1) = 37.834$, p -value <0.00001). This corresponded with the changes to foundational programming and also the range of content in first-year units.

The first-attempt pass rate for the foundational programming unit in the previous curriculum was 53%. In the renewed curriculum, it was also 53% (but it is worth reiterating that now the students with substantial prior learning in programming no longer undertake the foundational programming unit). There was a significant difference in the first-attempt pass rate for foundational programming between 2014 and 2016 when it was 43% and 2017 and 2018 when it rose to 66% ($\chi^2(1) = 31.772$, p -value <0.00001). This change corresponded to the period when there was a change in the assessment method in foundational programming.

The alternative entry point was a strategy to retain students that were already skilled in the area of programming. 107 BICT students have taken the direct route to the second introductory programming unit between 2014 and 2018; 70% passed on their first attempt. Of those students, 43% have graduated, and 36% are continuing. There is a significant difference between the students who have used this entry point, with only 21% of these students withdrawing, compared to the 44% of students overall that have withdrawn from the BICT ($\chi^2(1) = 14.35$, p -value = 0.00015).

6.1 Domestic Students

Figure 1(a) illustrates the number of commencing ICT domestic students as well as the rate of course attrition over the time span of this study. 2014 is an anomaly as it includes the students who commenced a previous ICT course and transferred to the new course, 49 of whom were domestic. If all students who transferred from the previous course are excluded, the total from 2010 to 2013 is 412 and from 2014 to 2017 (a similar 4-year period) minus the students that transferred is 322.

The domestic course attrition has significantly decreased with the total domestic withdrawn students from 2010 to 2012 at 64%, and from 2014 to 2016 (a similar 3-year period), it was 55% ($\chi^2(1) = 19.445$, p -value = 0.00006). This decline has continued for 2017 and 2018, with attrition rates of 38% and 20%, respectively.

There was a significant difference in the academic success of domestic students as shown by those that passed all their first-year units. From 2010 to 2012, 24% of domestic students passed all units compared to 49% for 2017–2018 ($\chi^2(1) = 23.562$, p -value <0.00001).

Many commencing domestic students have completed an ICT-related subject at year 11 or 12. These subjects are mostly computer science, information technology, or information systems. Of the 171 domestic students whose entry subjects are known in the BICT, 15% of them have not studied an ICT-related entry subject. It is not possible to do a comparison with the previous ICT courses, as the academic history of those students is not recorded in the current system. Of the students that entered the BICT without an ICT background, 48% have withdrawn, 12% have graduated, and the rest are continuing with their BICT. In comparison, of those that did have a year 11 or 12 ICT background, 30% have withdrawn, and 20% have graduated, with the remaining 50% still continuing.

The GCT major has been particularly attractive to domestic students, with 34% of domestic students enrolling in this major. For domestic students, both the GCT and SD majors have equivalent withdrawal rates of 22% and 24%, respectively ($\chi^2(1) = 0.327$, p -value = 0.547). They also have equivalent completion rates of 40% and 36%, respectively ($\chi^2(1) = 0.162$, p -value = 0.688).

6.2 International Students

Figure 1(b) illustrates the number of commencing ICT international students as well as the rate of course attrition. From 2010 to 2014, the commencing cohort was mostly domestic. The 2015 cohort was relatively balanced with 44% international students. Unfortunately, the qualifications of many of those students proved substandard to the entry requirements, and a large proportion of them withdrew by the end of their first semester. This had an impact on the 2016 international commencements as new entry evaluation procedures had to be established. By 2018, international students outnumbered domestic students. Removing the students that transferred from a previous course, there is a significant difference in the number of international students commencing from 2014 to 2017 in comparison to 2010–2013 ($\chi^2(1) = 29.7698$, p -value < 0.00001).

7 Discussion

This study provided evidence that the careers-focused strategy to attract more domestic students had merit, as evidenced by the addition of a games-career-focused major that attracted 34% of the domestic students. Had this option not been available and if these gamers opted to pursue their interest in games elsewhere, the impact on domestic commencements could have been catastrophic for UTAS and the local ICT industry.

One major concern is that overall Tasmanian ICT domestic commencements have dramatically declined. Some non-ICT students were attracted to the course but not enough to increase overall commencements to the required level to meet industry

demand for ICT graduates. Nationally there was a 126% increase between those two periods of time for commencing domestic ICT students [4]. So Tasmanian commencements are not following the national trend, possibly as an outcome of low enrolments in ICT year 11 or 12 subjects in Tasmania [2] or as a result of the size of the ICT industry in Tasmania [21].

There was a statistically significant difference in international student commencements, but it is important to consider what was happening nationally with international student commencements and acknowledge that UTAS is classed as a regional university and this has permanent residency visa implications which also influence an international student's choice of location. The BICT at UTAS has seen a 160% growth in international students from 2014 to 2017 compared to 2010–2013, while nationally there has only been a 120% growth between those two time periods [4], indicating the growth is above the national average over the evaluation period. Thus, while the new curriculum was not associated with better ICT domestic student commencement rates, there is an association with better ICT international student commencement rates.

Further research into the influence of career-based pathways is warranted based on the results for the games-career-focused major. To attract more students, the career approach was utilized again as part of a 2018 ICT curriculum review undertaken at the University of Tasmania. After extensive consultation with the local ICT industry, more specialist career outcomes were identified; there was significant demand for security specialists, data scientists, and software designer/developers. There was enough demand to retain the game designer/developer and business analyst roles. The skill sets for these roles were identified using SFIA, and five career-focused majors were offered in 2019. An empirical evaluation of this revised curricula will be conducted in a few years to analyze the changes. The total domestic student commencements for 2019 have improved on 2018, but as there will be yearly fluctuations, a longer-term study is required before drawing any conclusions.

There has been a significant improvement in the academic success of both domestic and international students. There were a few changes that could have influenced the first-year pass rates, foundational programming pass rates, and graduation rates:

- The broader first-year curriculum
- The change to the teaching and assessment style for foundational programming
- The alternative entry point for students with prior programming experience

The fact that a student passes foundational and/or introductory programming is a huge incentive to persist with the course, though 33% of domestic students compared to 6% of international students with prior programming withdrew. In a separate study, an investigation into when the students were withdrawing found that domestic students were more likely to withdraw in the latter years of their degree than international students [28]. It was theorized that domestic students have more freedom to find employment, and this leads to students changing to part-time or even withdrawing when they have enough skills to find employment [28]; this theory needs further investigation.

This study provided statistically significant evidence of a decline in domestic course attrition from 2010 to 2013 in comparison to 2014–2018. An adverse intake of international students in 2015 is confusing the analysis of the overall changes to international student course attrition rates though they too are tending toward an improvement. Key changes during this period were the strategies to improve student perceptions, motivation, engagement, and academic success which have arguably created this difference. However, as already discussed in the methods section, during the same period, there were also a number of uncontrolled changes. As the course changes were not independent of each other or independent from these uncontrolled changes, it is not possible to identify if one change was more successful than another in reducing course attrition.

8 Conclusion and Future Work

The purpose of this chapter was to address the research question: What is the impact of strategies designed to amend misconceptions and improve perceptions, motivation, engagement, and academic success on the quantity of graduates? This chapter has provided a comprehensive evaluation over a significant time period of various strategies that were employed throughout a curriculum and impacted on a student's entire learning experience. There was evidence to support the assertion that a career-focused approach to curriculum design could increase course commencements. There was strong evidence to support the assertion that employing strategies curriculum-wide to amend misconceptions and improve perceptions, motivation, engagement, and academic success can significantly reduce course attrition. This chapter contributes to the field of ICT curriculum design as it provides practical implementation techniques for strategies that have been shown to have positive long-term outcomes.

When combined with the concurrent study that provides strong evidence that the technical skills have been maintained, and that the professional skills have improved [24], these studies have shown it is possible to increase the quantity of competent graduates which will positively impact on the growth of the ICT industry.

References

1. ACS Statistical Compendium. 2012. <http://www.acs.org.au/news-and-media/news-and-media-releases/2012/acs-statistical-compendium-2012>. Accessed 17 Aug 2013
2. Australian Information Industry Association (AIIA). 2012. ICT skills and training development: A 'State of Play' paper. https://www.aiaa.com.au/influence-And-leadership/policy-submissions/submissions/policies-and-submissions/2012/ICT_skills_and_training_development_23_11_2012.pdf. Accessed 14 Sept 2019
3. L. Carter. 2006. Why students with an apparent aptitude for computer science don't choose to major in computer science, in SIGCSE '06. 1–5 Mar 2006, Houston, TX, USA, 27–31

4. Australian Government: Department of Education: Student Data, <https://www.education.gov.au/student-data>. Accessed 14 Sept 2019
5. M. Butler, M. Morgan, J. Sheard, K.F. Simon, A. Weerasinghe, Initiatives to increase engagement in first-year ICT, in *Proceedings of the 2015 ACM Conference on Innovation and Technology in Computer Science Education*, (ACM, New York, 2015), pp. 308–313
6. A. Petersen, M. Craig, J. Campbell, A. Taffioovich, Revisiting why students drop CS1, in *Proceedings of the 16th Koli Calling International Conference on Computing Education Research*, (ACM, New York, 2016), pp. 71–80
7. M. Haungs, C. Clark, J. Clements, D. Janzen. 2012. Improving first-year success and retention through interest-based CS0 courses, in Proceedings of the 43rd ACM Technical Symposium on Computer Science Education, Raleigh, NC, USA
8. V. Smith, E. Fernandez. 2014. Factors affecting first year retention of CIT students, in Proceedings of the ASEE Annual Conference & Exposition: 1–11. Academic Search Ultimate, EBSCOhost
9. Deloitte Access Economics. 2019. Australia’s digital pulse 2019: Booming today, but how can we sustain digital workforce growth?, <https://www.acs.org.au/content/dam/acs/acs-publications/Digital-Pulse-2019-FINAL-Web.pdf>. Accessed 14 Sept 2019
10. T. Beaubouef, J. Mason, Why the high attrition rate for computer science students: Some thoughts and observations. *ACM SIGCSE Bull.* **37**, 103–106 (2005)
11. T. Beaubouef, P. McDowell, Computer science: Student myths and misconceptions. *J. Comput. Sci. Coll.* **23**(6), 43–48 (2008)
12. M. Biggers, A. Brauer, T. Yilmaz, Student perceptions of computer science: A retention study comparing graduating seniors with cs leavers, in *Proceedings of the 39th SIGCSE Technical Symposium on Computer Science Education (SIGCSE '08)*, (ACM, New York, 2008), pp. 402–406
13. M. Morgan, M. Butler, J. Sinclair, C. Gonsalvez, N. Thota, Contrasting CS student and academic perspectives and experiences of student engagement, in *Proceedings Companion of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE 2018 Companion)*, (ACM, New York, 2018), pp. 1–35
14. S. Ragsdale, Pursuing and finishing an undergraduate computing course: Insights from women computing graduates. *J. Comput. Sci. Coll.* **30**, 5 (2015)
15. M. Säde, R. Suviste, P. Luik, E. Tõnisson, M. Lepp, Factors that influence students’ motivation and perception of studying computer science, in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, (ACM, New York, 2019), pp. 873–878
16. J. Sheard, A. Carbone, A. Hurst, Student engagement in first year of an ICT course: Staff and student perceptions. *Comput. Sci. Educ.* **20**(1), 1–16 (2010)
17. A. Kapoor, C. Gardner-McCune, Considerations for switching: Exploring factors behind CS students’ desire to leave a CS major, in *Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education*, (ACM, New York, 2018), pp. 290–295
18. M.R.H. Roberts, T.J. McGill, P.N. Hyland, Attrition from Australian ICT courses: why women leave, in Proceedings of the Fourteenth Australasian Computing Education Conference, pp. 15–24, Melbourne, Australia, 2012
19. C. Lang, J. McKay, S. Lewis, Seven factors that influence ICT student achievement. *SIGCSE Bull.* **9**(3), 221–225 (2007)
20. L. Barker, C. McDowell, K. Kalahar, Exploring factors that influence computer science introductory course students to persist in the major, in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education (SIGCSE '09)*, (ACM, New York, 2009), pp. 153–157
21. Tasmania: Department of State Growth. 2014. Information and communication technology: Sector Summary 2014. https://www.stategrowth.tas.gov.au/__data/assets/pdf_file/0006/89583/ICT.pdf. Accessed 14 Sept 2019
22. N. Herbert, K. de Salas, I. Lewis, J. Dermoudy, L. Ellis, ICT curriculum and course structure: The great balancing act, in *Proceedings of the Sixteenth Australasian Computing Education Conference-Volume 148*, (Australian Computer Society, Inc, 2014), pp. 21–30

23. ACM, Association for Computing Machinery curricula recommendations. 2019. Retrieved 8 Oct 2019, from <http://www.acm.org/education/curricula-recommendations>
24. N. Herbert, D. Herbert, T. Acuna, K. de Salas, E. Wapstra, Integrating the development of professional skills throughout an ICT curriculum improves a graduate's competency, in 2020 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, 2020
25. N. Herbert, K. de Salas, I. Lewis, M. Cameron-Jones, W. Chinthammit, J. Dermoudy, L. Ellis, M. Springer. 2013. Identifying career outcomes as the first step in ICT curricula development, in Fifteenth Australasian Computing Education Conference, Adelaide, Australia
26. N. Herbert, Impact of student engagement on first year ICT performance, in International Conference on Computational Science and Computational Intelligence, Las Vegas, NV, 2017, pp. 1085–1090
27. I.O. Pappas, M.N. Giannakos, L. Jaccheri, Investigating factors influencing students' intention to dropout computer science studies, in *Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education*, (ACM, New York, 2016), pp. 198–203
28. N. Herbert, D. Herbert, T. Acuna, K. de Salas, E. Wapstra, An exploratory study of factors affecting attrition within an ICT course, in *22nd Australasian Computing Education Conference*, (ACM, New York, 2020)

Incorporating Computer Programming into Mathematics Curricula to Enhance Learning for Low-Performing, Underserved Students



Alan Shaw and William Crombie

1 Introduction

By some estimates, as many as two thirds of American adults currently suffer from some type of math phobia due to bad educational experiences with mathematics [1]. Our work with students performing in the bottom quartile in mathematics tests has demonstrated some ways that adding technologies like mobile apps with a particular pedagogical approach can help. Many studies have shown that targeted use of multimedia technologies can make a significant impact on a student's sense of ownership and engagement [2–6], and yet the lowest-performing students are often the most disengaged, while they are often in a resource-poor environment, with less frequent access to rich interactive technologies. The low cost, the prevalence, and the social appeal of tablets in the classroom can help.

In four schools in Atlanta where our research was conducted, more than 95% of the students are eligible for free and reduced cost lunch, and they only had access to computer labs on the average of once a week, instead of the daily access that is available in more affluent schools. And yet, most of the students we worked with either owned or had some type of access to cellphones. With this in mind, we applied for and received an NSF Early-Concept Grant for Exploratory Research (EAGER) in which we examined the feasibility of combining mathematics curricula with simple mobile app development [7].

Throughout the fall of 2016, and the spring and summer of 2017, we have undertaken to develop a new approach to using mobile apps for introducing

A. Shaw (✉)

Computer Science, Kennesaw State University, Kennesaw, GA, USA

e-mail: ashaw8@kennesaw.edu

W. Crombie

The Algebra Project, Cambridge, MA, USA

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_2

educational mathematics simulations in a set of middle-school math classes in a way that blends computing ideas with math instruction. The curricular material involved was developed specifically for students in the lower quartile by the Algebra Project, to further enhance these students' conceptual and procedural knowledge of mathematics content and to strengthen students' mathematical practices. The Algebra Project curriculum is based on an experiential mathematics pedagogy and a curricular process with extensive past documentation and research and evidence that it assists the low-performing students that we were targeting [8–13].

2 Study Design

The experientially based curriculum of the Algebra Project engages students in concrete events and activities that are then examined reflectively to analyze them mathematically. This reflection and exploration happens during the following five-step curricular process that occurs during the many different units of the Algebra Project curriculum:

- Step 1: Experience a physical event as a group.
- Step 2: Represent that event through drawings or by creating models.
- Step 3: Describe the event informally and intuitively, using natural and idiomatic language.
- Step 4: Translate the idiomatic description into a structured, formal, feature-rich description.
- Step 5: Create a symbolic representation of the event using mathematical formalisms.

For our EAGER grant, we proposed integrating basic programming experiences involved in developing mobile apps into these five steps in the following way:

- Step 1: Experience a simulation of the physical event.
- Step 2: Represent that simulation through drawings or by creating models.
- Step 3: Describe the simulation informally and intuitively, using natural and idiomatic language.
- Step 4: Translate the idiomatic description into a structured set of visual program blocks that represent functional units.
- Step 5: Connect the functional programming units to recreate the simulated experience as a mobile app.

Two units of the Algebra Project curriculum were chosen for this intervention: the Road Coloring module and the Race Against Time module.

3 The Curricular Units

The first unit of study in this research was the Road Coloring module. In this unit, functions are modeled as simultaneous physical movements by groups of students, and students use ordered pairs and point on a coordinate plane and arrow diagrams to represent the functional transitions. The notions of domain and range are developed and have easily accessible, concrete interpretations.

Along with the conceptual underpinning of the function concept, students are introduced to the Road Coloring challenge based on a famous problem in theoretical computer science first stated in a paper by Adler, Goodwyn, and Weiss [14] that remained unsolved for over 30 years. The original problem asks if all strongly connected, aperiodic, directed graphs have an edge labeling for which a synchronizing instruction exists. In this unit, the directed graphs become “cities” that students represent with the points on the floor (the vertices) serving as “buildings” with numbered “addresses” (building 1, building 2, etc.). And the paths between the points serve as the edges and as one-way roads. The students then attempt to find a set of directions that will get everyone from their different vertices to the same building at the same time.

Once the students physically experience the concept of functions in this manner, multiple representations are introduced. As an example, below are arrow diagrams the students produce of a city with three buildings, with one red road and one blue road leading away from each building (this representation constitutes an edge-colored directed graph with three vertices) (Fig. 1).

Other standard representations are also introduced to the students, and included are representations of 0–1 stochastic matrices (such as permutation matrices) and one-out directed graphs. The last two representations represent important innovations for function representations in the high school curriculum and were particularly important in the mathematics research that led to the eventual solution of the original Road Coloring problem [15–17].

The second unit of study in this research was the Race Against Time module. This module uses relay races as the shared concrete event, and through this event, students are introduced to the concept of linear equations that ultimately lead to the form $ax + b = c$. Linear equations are developed within the context of “detective work” to determine the locations a team has visited in the course of a race. The concept of slope is introduced and analyzed, as students graph relay race trajectories

Fig. 1 Road Coloring task



as “distance traveled” versus time elapsed. The “average velocity” of a relay race leg allows the natural introduction of slope.

The relay races are not outdoor foot races. Students race by stacking plastic cubes on top of one another in a fixed amount of time. The cubes are then laid out in a direction determined by a flip of a coin from a starting point, or from the last endpoint achieved by a previous student racing for the same team. As each team member adds a new displacement during the race to a set of previous displacements, a new distance from the origin is produced. Students produce tables, arrow diagrams, graphs, and equations of the resulting displacements and use these to explore various linear transformations involved in their analyses (Fig. 2):

This specific development of linear equations makes direct contact with the more general development of the concept of function in the Algebra Project Road Coloring module. The two approaches, from Road Coloring to Race Against Time, provide students with a binocular and complementary perspective on these central concepts of early algebra.

3.1 Developing the Simulated Units: Road Coloring

The Road Coloring mobile app was designed in the fall semester of the 2016–2017 school year to simulate the creation of Road Coloring “cities” made up of a directed graph of vertices called “buildings” connected with edges called red and blue “roads.” The app was originally developed using MIT App Inventor and its Visual Blocks system, and it was tested in Algebra Project classrooms in San Francisco in the fall of 2016, and then it was further developed and tested in the four Algebra Project classrooms in Atlanta in the spring and summer of 2017.

Without the app, constructing cities of more than four buildings (vertices) was very difficult for the students, but with the app, students were able to come up with synchronizing instructions for cities of more than ten buildings. The following shows cities of three buildings, four buildings, and ten buildings, respectively (Fig. 3):

Initially the students used the visual programming block system from MIT App Inventor, to create the apps, but it ran much too slowly when a class of 20+ students were using the web-based program at the same time. Because of this, we developed a scaled down version of the same visual programming block system that worked with blocks that were specific to only the Road Coloring and the Race Against Time

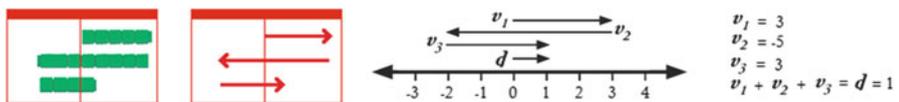


Fig. 2 Various Race Against Time representations

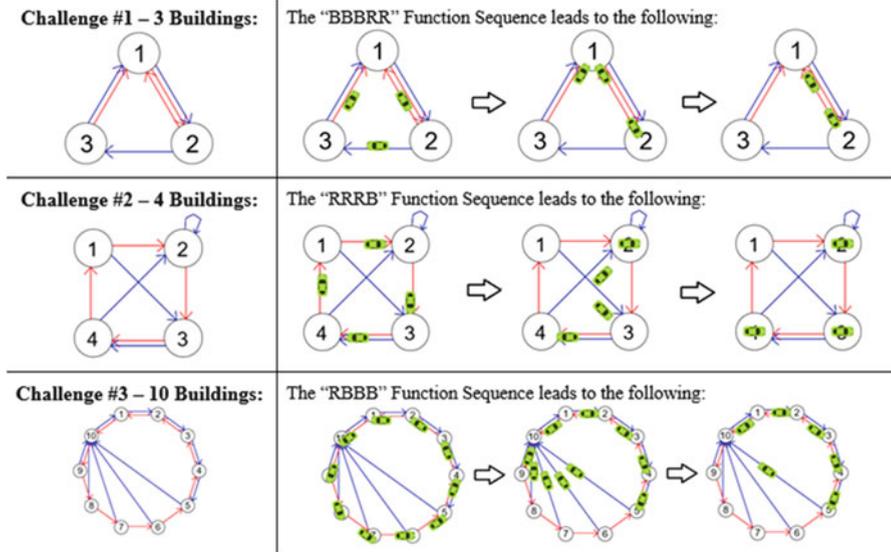


Fig. 3 Road Coloring cities and synchronizing instructions

Fig. 4 Road Coloring program blocks

This screen shows what the blocks on the right produce.

```

call CreateCity
  CityTypeInput: call SetCityType
  Has3BuildingsInput: true
  Has4BuildingsInput: false
  Building1RoadsInput: call SetBuilding1Roads
    RedGoesToInput: BlueGoesToInput
  Building2RoadsInput: call SetBuilding2Roads
    RedGoesToInput: BlueGoesToInput
  Building3RoadsInput: call SetBuilding3Roads
    RedGoesToInput: BlueGoesToInput
  Building4RoadsInput:
  
```

units, unlike App Inventor which allows students to create blocks for more general purposes. The blocks for a three-building city appeared as follows (Fig. 4):

3.2 Developing the Simulated Units: Race Against Time

The Race Against Time app was designed in the spring semester of the 2016–2017 school year to simulate the relay races that make up the Algebra Project’s Race Against Time unit. The app is also an Android app, and it was tested in Algebra Project classrooms in Atlanta in the spring and summer of 2017.

The Relay Race simulation consisted of students dragging cubes that were scrolling along in a box at the top of the screen and stacking them in vertical line. The challenge had different levels of difficulty because of different time constraints placed on them that the students were able to control programmatically (Fig. 5):

After finishing the race, the arrow diagrams produced by the app look the same as the textbook arrow diagrams that the students are familiar with in the classroom. Yet in the app, the students can modify the magnitude and direction of the vectors in real time. In the following diagram is a set of visual programming blocks the students could have constructed to create the simulation shown above (Fig. 6):

Using the app, students determine how many legs they will have in their race, which can be any number from 2 to 10, and they also determine how many seconds they will have to stack cubes during each leg, which can be some number from 2 to 25. The students run a simulation where a group of cubes moves along the top of the screen horizontally, and the students must drag them one by one down onto a stack that they are creating. When a leg ends, their stack has a certain amount of cubes which represents a magnitude. And the students press a button that randomly assigns a left or right direction for their stack, which gives it both a magnitude and a direction, making it a vector.

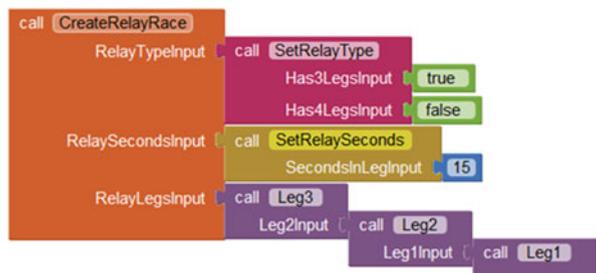
Once a student has finished all of their legs, they have a set of consecutive vectors that have a total set of magnitudes (called the “total distance”) and a resulting displacement from the origin (called the “total displacement”). The total distance concept involves adding the absolute value of the distance traveled during each leg,

Fig. 5 Race Against Time simulation



Seconds Left: 5
Blocks Stacked: 7

Fig. 6 Race Against Time programming blocks



but it also just represents how many cubes the students stacked during all of the legs combined. The total displacement represents the final distance and direction from the origin, so it represents a new vector that is the result of adding up all of the vectors from each leg of the race.

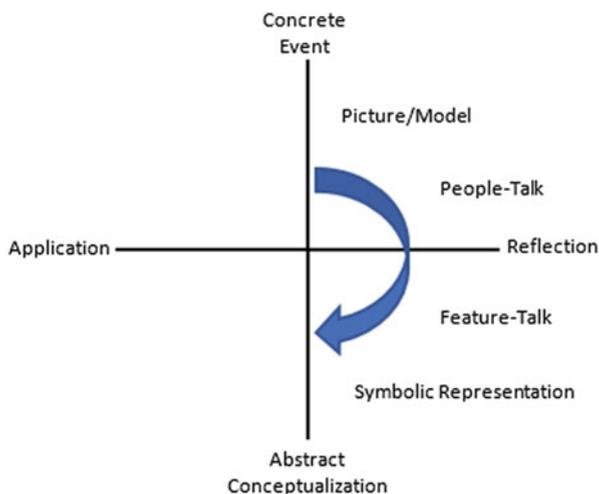
After finishing all of the legs, students calculate the total distance and the total displacement, and then the students examine tables and arrow diagrams representing the details of their race in different ways. Then the students use the app to solve linear equations that involve the resulting displacements that occur when legs of their race are modified using various linear transformations.

4 Theoretical Underpinnings

The five-step curricular process that we have adapted for this research exploits an experiential learning cycle that begins with the students working through a concrete event and moving progressively to an abstract symbolic and mathematical representation of that event. During this process, the students reflect upon the event by identifying important features captured in informal language (called people-talk) and formal representations (called feature-talk) of the event to figure out how the features are related with symbolic representations (Fig. 7).

The multiple representations of the event that are constructed as students move through the experiential learning cycle are described by W. V. Quine’s notion of the language foundations of mathematics as a circular curricular process [18]. Quine saw mathematics as a conceptual language that has its beginnings in the structuring and regimentation of ordinary discourse. In the curricular process, this structuring occurs with the students in their discussions about their exploration of the important

Fig. 7 Experiential learning cycle



features of the event. In this discourse, students try to make sense of abstract symbolic representations of conventional mathematics. Students work to gain the ability to read algebraic sentences in a meaningful and interpretive fashion. This ability to interpret the symbols of mathematics enables students to affect a shift from algorithms and computation to logic and reasoning as the basis for problem-solving in mathematics.

Our research has allowed us to extend that process of reflection through the interaction with simulations and computing. By developing apps which produce simulations of the concrete experiences being used in the Algebra Project curriculum, students acquire an additional dimension to the experiential learning process they are engaged in. There are both cognitive and affective dimensions to the use of app-based simulations.

The work of Jerome Bruner is particularly relevant here. Bruner has asserted the importance of representation in the learning of knowledge domains in general and of mathematics in particular. In this work, *Toward a Theory of Instruction* [19], Bruner describes the structure of a knowledge domain in terms of the representations that are used to capture its content. This representational view captures the structure in terms of the modes of representation (enactive, iconic/graphic, or symbolic), the economy of the representation (the cognitive load that students are required to carry), and the power of the representation (descriptive, explanatory, and predictive), and we would add to Bruner's list the scope of the representation (the degree to which it facilitates near transfer to problems/questions within its defining context or far transfer to problems/questions outside of its original/defining context).

For example, students only used the Road Coloring app simulation after they had built a model city in their classroom. The app's simulation was always a representation for the students of the real event. The app's simulation also introduced a hybridization of Bruner's modes of representations. In traditional mathematics, classroom students typically use enactive representations of mathematical concepts. We call them manipulatives. And students construct iconic representations of mathematics concepts: pictures, graphs, and diagrams. The app simulations provided students with enactive-iconic representations of the mathematical concepts they were engaged with. This type of enactive-iconic representation created a space of possibilities for different types of student engagement and understanding in the classroom. The enactive-iconic representation gave students capability to rapidly construct and manipulate cities of greater complexity than could easily be made of real materials or from paper and pencil.

We also note that in the case of the Relay Race app simulation, the enactive-iconic representation provided by the app gave students the cognitive space to apply visual reasoning and logic to the solution of linear equations. This suggests that the traditional learning progression that takes students through one-step, two-step, to multi-step linear equations may be more a consequence of a didactic choice than a requirement of the cognitive stages that students must go through to achieve mastery of the subject.

One final point is worth making. The concrete events of the Algebra Project's Curricular Process are in stark contrast to the "hands-on" paradigm employed in

many math curricula. In many of these curricula, every lesson can have its own hands-on component, each one separate and distinct, characterizing the particular lesson. Over the years, the Algebra Project has targeted what they consider to be a few foundational events/experiences which meet the representational criteria given above of having broad scope. These experiences embody representations of concepts that apply to broad swaths of the mathematical landscape primarily at the level of introductory algebra. These events and their representations thus act as grounding metaphors for the construction of the foundational concepts of algebra.

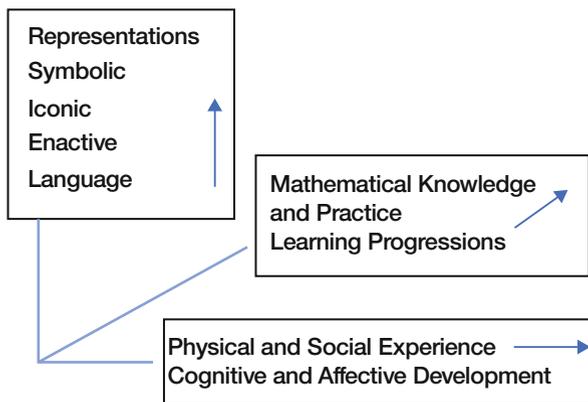
Our work with app simulations for two of these grounding metaphors suggests that by providing students, particularly “low-performing” students, with enactive-iconic representations, we create a space for them to apply cognitive abilities that do not normally present in the traditional mathematics classroom. We have yet to see, given the time and the affordances which this technology provides, if the traditional learning progressions will need to be re-written especially for what are now considered “low-performing” students.

These considerations are what lead us to develop our app simulations according to a three-dimensional approach. We sought to design them to include (1) a physical experience that could be shared as a social activity with mathematical implications; (2) enactive-iconic representational objects that can connect naturally to a discourse about the experience and activity; and (3) mathematics knowledge and computational logic that is involved in specific learning unit/learning progression currently in the classroom. Our design goals for app simulations of this nature are thus given by the following image (Fig. 8):

5 Initial Results

Over 200 students participated in some aspect of this intervention. Initially, we had intended to focus on just four schools in the Atlanta area (Brown, Bunche, Harper

Fig. 8 Three-dimensional design of app simulations



Archer, and Long), but due to administrative problems, we were not able to begin the work in the Atlanta schools until the spring of 2017. So in the fall of 2016, we worked with approximately 50 students from the June Jordan School for Equity in San Francisco. During the spring of 2017, we finally began work with about 140 students from the four Atlanta schools. And during the summer of 2017, we worked with about 15 more students as part of a summer school program at the Harper Archer Middle School in Atlanta.

In our work during the fall and spring, our instructional material and our modified visual programming block system were going through an incremental development process, whereby we would try a particular version of the material and then make modifications based on student and teacher feedback or our own analysis. This led to our having a well-defined set of materials and a working version of our own modified visual programming block system by the summer of 2017.

The difference between the work in the summer and the work in that occurred earlier in the fall and spring was that during the fall and spring, our materials were going through constant revision, making it difficult to evaluate formally the impact of any one set of materials. In the summer, however, our materials development process was finished, and so we worked with a stable set of materials during the entire 4 weeks of the summer school. It was during this time that we were able to implement a set of pre- and post-test to begin to evaluate how the work we did contributed to a sense of ownership, engagement, and comprehension within the students.

Our students who participated in the work and the pre- and post-tests over the summer were a very small sample, only 15 students. But all 15 of the students said in one-on-one interviews we conducted that they enjoyed working with the apps, and some explained that they thought of it as a “hands-on” way of doing math. Some went on to say that they felt a true sense of accomplishment when they were able to solve difficult problems using the simulations that would have been much more difficult with only pencil and paper. In many cases, we recorded students showing other students the solutions they came up with displaying a sense of accomplishment in their work, and one student said he was doing this to prove to the other students how “smart” he was.

The summer students were all rising sixth graders, entering the seventh grade, and none of them had been in an algebra class yet. Therefore, on the pre- and post-test, we had the students work on algebraic problems before and after they worked on the curricular material involving the apps. The students spent 2 weeks working on the Road Coloring unit and 2 weeks working on the Race Against Time unit.

The Road Coloring unit dealt with how Red and Blue functions (enactively and iconically represented by the roads) which take inputs and produce an output, and how a sequence of those functions, which is function composition, can be used to produce a particular output. In the pre- and post-tests, students were asked to analyze a set of connected cell towers and how phone calls are routed between them to get to a particular target. Before the unit, only 1 of the 15 could correctly draw composed sequence of routes and solve problems to achieve a correct route for certain calls. After the unit, eight students could accomplish the same tasks.

The Race Against Time unit dealt with representing vectors with arrow diagrams and using the arrow diagrams to solve linear equations. In the pre- and post-tests, students were asked to represent trips on the Marta (Atlanta's tramline) using arrow diagrams and, when given a sequence of trips, to determine the total distance traveled and the ultimate displacement (the final distance from the beginning of the trip to the end of the trip). On these tasks, only three students could draw an accurate arrow diagram representing a complex trip, only seven students could calculate the correct total distance traveled, and only two calculated the correct displacement. After the unit, 11 students were able to draw an accurate arrow diagram, ten were able to calculate the correct distance, and eight were able to calculate the correct displacement.

Our tests were designed to show increases in conceptual understanding, visual reasoning, and representational logic, not just the ability to manipulating math symbols when solving linear equations. And the majority of the students in both units did improve in these areas. However, we also did do some tests of the ability of students to manipulate mathematical symbols as well after the second unit. Students were shown the following arrow diagram, which was similar to the ones they dealt with in the app simulation (Fig. 9):

And without any instruction in algebra, students were then shown the following equation and asked to solve for X :

$$4 + X + 4 = 13$$

Fourteen of the students were able to answer the question correctly. Then a diagram was shown for the equation $4 + X + 4 = -13$, and only ten of the students were still able to solve for X . Then the following was shown with the equation $4 + 2X + 4 = -2$, and still ten students (67%) solved this expression for X , again using non-algebraic processes (Fig. 10).

This indicated that for those ten students, the conceptual understanding of what that equation represents was grounded in visual reasoning and representational logic. From many indicators, we saw obvious improvements in ownership, engagement, and increased comprehension.

Fig. 9 Displacement arrow diagram

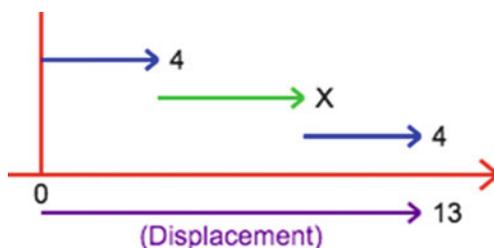
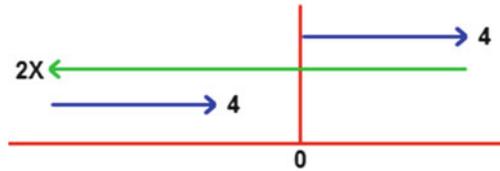


Fig. 10 Another displacement arrow diagram



6 Conclusion and Future Directions

Because our research was part of an EAGER grant, which involves early-concept exploratory research, our work was primarily focused on demonstrating that a fuller study is warranted. Although about 200 students participated in some aspect of the research, only 15 students in the summer were part of the sample that dealt with the actual pre- and post-test. The others helped in our exploratory and developmental effort that the EAGER grant is designed for. However, the summer students didn't have a traditional math class context where we could spend more time with them and see how they performed on traditional mathematics materials before and after our intervention. We see this as the logical next step for our research, as well as to expand on the number of curricular units that we develop simulations for using our three-dimensional design criteria.

And finally, we also see the potential for expanding this intervention into science curricula as well. The Algebra Project was a great fit for this intervention because simulations and experiential learning can be aligned by identifying an appropriate simulation for the concrete experiences in their curriculum. Other math curricula may not always have easily identified concrete experiences that are enactive-iconic and reasonable to simulate. However, we believe that science classes are often focused on physical phenomena that would satisfy this criterion and be reasonable to simulate, and therefore this presents us with another area for future research.

References

1. M. Burns, *Math: Facing an American Phobia* (Math Solutions, Sausalito, 1998)
2. J.S. Brown, Growing up digital: How the Web changes work, education, and the ways people learn. *Change*, 10–20 Mar/Apr 2000. Also accessible at USDLA Journal, 6(2) Feb 2002. http://www.usdla.org/html/journal/FEB02_Issue/article01.html
3. J. Parsons, P. McRae, L. Taylor, *Celebrating School Improvement: Six Lessons from Alberta's AISI Projects* (School Improvement Press, Edmonton, 2006)
4. R.B. Kvavik, J.B. Caruso, G. Morgan, *ECAR Study of Students and Information Technology 2004: Convenience, Connection, and Control* (EDUCAUSE Center for Applied Research, Boulder, 2004), p. 784. *Br. J. Educ. Technol.* 39(5)
5. Project Tomorrow, Unleashing the future: Educators "speak up" about the use of emerging technologies for learning. *Speak Up 2009 National Findings. Teachers, Aspiring Teachers & Administrators*, May 2010. Retrieved Dec 2010 from www.tomorrow.org/speakup/

6. K. Barnes, R. Marateo, S.P. Ferris, Teaching and learning with the net generation. *Innov. J. Online Educ.* **3**(4) (2007). Reprinted in The Fischler School of Education and Human Services at Nova Southeastern University; Pennsylvania
7. A. Shaw, K. Hoganson, W. Crombie, Incorporating computer programming into middle school mathematics curricula to enhance learning for low performing, underserved students (NSF/EAGER#1651092). Kennesaw State University Research and Service Foundation
8. M.M. West, *Final Report: The Development of Student Cohorts for the Enhancement of Mathematical Literacy in Under Served Populations (NSF/DRK12#0822175)* (Algebra Project, Inc., Cambridge, 2015)
9. E. Dubinsky, R.T. Wilson, High school students' understanding of the function concept. *J. Math. Behav.* **32**, 83–101 (2013)
10. D.N. Brewley-Corbin, Case Study Analysis of Mathematics Literacy Workers' Identity and Understanding of Numbers Within a Community of Practice, Doctoral dissertation, University of Georgia, 2009
11. M. Gresafi, T. Martin, V. Hand, J. Greeno, Constructing competence: A analysis of student participation in the activity systems of mathematics classrooms. *Educ. Stud. Math.* **70**, 49–71 (2009)
12. M.M. West, F.E. Davis, M. Currell, *Algebra for all in Grade 8: a longitudinal study of mathematics reform at Dr. M.L.King Academic Middle School, San Francisco* (Program Evaluation & Research Group, Lesley University, Cambridge, 2006)
13. M.M. West, F.E. Davis, *The Algebra Project at Lanier High School, Jackson, MS* (Program Evaluation & Research Group, Lesley University, Cambridge, 2004)
14. R.L. Adler, L.W. Goodwyn, B. Weiss, Equivalence of topological Markov shifts. *Israel J. Math.* **27**(1), 49–63 (1997)
15. G. Budzan, A. Mukherjea, A semigroup approach to the road coloring problem, in *Contemporary Mathematics*, vol. 261, (AMS, Providence, 2000), pp. 195–207
16. G. Budzan, Semigroups and the generalized road coloring problem. *Semigroup Forum* **69**, 201–208 (2004)
17. A. Trahman, The road coloring problem. *Israel J. Math.* Vol. **172**, 51–60 (2009)
18. W.V. Quine, *Mathematical Logic* (Harvard University Press, Cambridge, 1982)
19. J. Bruner, *Toward a Theory of Instruction* (Harvard University Press, Cambridge, 1974)

Examining the Influence of Participating in a Cyber Defense Track on Students' Cybersecurity Knowledge, Awareness, and Career Choices



Michelle Peters, T. Andrew Yang, Wei Wei, Kewei Sha, and Sadegh Davari

1 Introduction

Countering cyber threats and protecting the cyber space are considered daunting tasks by most organizations and businesses. The constantly evolving nature of cyber threats makes traditional, passive control mechanisms such as checklists ineffective; an organization's successful compliance to cybersecurity regulations and guidelines may actually create a false sense of security. Furthermore, the proliferation of various cyber defense devices such as firewalls and security appliances may result in a complicated infrastructure, which reduces network visibility and may actually harm an organization's effort in effectively detecting and mitigating cyber threats.

According to the Center for Cyber Safety and Education, unfilled cybersecurity jobs are expected to reach 1.8 million by 2022 [3]. "68% of workers in North America believe this workforce shortage is due to a lack of qualified personnel" [3]. Similar to the challenges the nation has faced in the past, threats to the cyber space and the necessary steps to mitigate those threats present both a crisis and an opportunity. Lack of skills and technical knowledge has been identified as the biggest barrier to successfully implementing cyber defense; this applies to both organizations and the nation as a whole. Higher-education institutions across the nation ought to take this new challenge and opportunity to modernize their computing degree programs in order to help the nation's response to these challenges, by preparing cyber-aware professionals to meet the nation's increasing demand for cybersecurity talents.

As suggested in the Federal Cybersecurity Workforce Strategy by the White House [1], one of the key initiatives was to "... collaborate with academic insti-

M. Peters (✉) · T. A. Yang · W. Wei · K. Sha · S. Davari
University of Houston, Houston, TX, USA
e-mail: petersm@uhcl.edu; yang@uhcl.edu; wei@uhcl.edu; sha@uhcl.edu; davari@uhcl.edu

tutions to develop guidance for cybersecurity core curriculum and allow colleges and universities to expand their course offerings.” What is left to be identified is what courses are to be offered at colleges and universities to prepare our graduates effectively based on what the market really needs. In a radio talk given by Allan Paller [5], founder and research director of the SANS Institute, he took a long-term view of our cybersecurity preparedness and pointed out that all sectors, especially government, are in desperate needs of cybersecurity professionals who can “do the technical things” such as security coding, penetration testing, and network forensics; those programs that only offer survey courses can only produce “admirers” rather than “fixers” of our problems.

What we need in our cybersecurity education programs are solid foundation knowledge and advanced hands-on skills. A competent cybersecurity practitioner should have fundamental understanding of computing and mathematics, and they should also be proficient with programming and problem-solving, all of which are already addressed in a Computer Science undergraduate program with quality. However, as reported in *Forbes Innovation*, out of the top 50 undergraduate CS programs in the USA, only 42% offer three or more information security-specific courses [4]. The percentage of programs that offer any specialization or track related to cybersecurity is even smaller. Therefore, a question left for us educators to answer is how we can strengthen existing CS programs to make substantial contribution to solving the cybersecurity talent crisis.

At University of Houston-Clear Lake (UHCL), we proposed to revamp the existing CS program to house a potential cybersecurity program. The advantages of this approach are as follows: (a) leverage existing curricular components to reduce the cost; (b) increase potential enrollment to the new program because CS students are the ones who are most likely to be interested in and qualified for it; and (c) make it possible for students to graduate from the new program within the credit hour limits mandated by the State, by weaving new program components with existing ones. The new courses and course modules are to be added to the existing CS curriculum, aiming to (a) compensate current deficiencies, (b) streamline cybersecurity-related content and offering, and (c) augment the program with more labs.

Our ultimate goal of revamping the Computer Science (CS) curriculum is to incorporate cybersecurity throughout the core courses, starting from CS1, in an organized way, so that the components in subsequent courses are built on the knowledge and training acquired in the previous courses. By continuously incorporating new findings into the existing cybersecurity components, we will be producing cybersecurity-aware CS graduates in the foreseeable future. The result could become a model for other institutions to follow.

In order to reach that goal, our near-term objective was to create a Cyber Defense Track in the existent Computer Science undergraduate degree program, by using the National Center of Academic Excellence in Cyber Defense in Four-Year Baccalaureate Education (CAE-CDE 4Y) as the model framework. The CAE-CDE is a program co-sponsored by the National Security Agency (NSA) and the Department of Homeland Security (DHS); its goal is to reduce vulnerability in

our national information infrastructure by promoting higher education and research in cyber defense and producing professionals with cyber defense expertise for the nation. Earning this designation is a rigorous process, and the requirements have been clearly stated.

In a preliminary self-study, we had identified gaps between our Computer Science undergraduate curriculum and the CAE-CDE 4Y requirements, including missing knowledge units (KUs) that need to be integrated into our existing CS degree program. To close the identified gaps in our CS curriculum, we have developed three brand new courses: Cyber Attack and Defense (CAD), Network Defense (ND), and Network Forensics (NF). Those courses were developed and offered in the 2017–2019 academic semesters. As a result, the focus of this chapter is to share the findings in regard to these recently developed courses.

2 Development of New Courses for the Cyber Defense Track

As stated above, in order to close the identified gaps in the existent Computer Science program, we developed three new courses, including Cyber Attacks and Defense, Network Defense, and Network Forensics. The course design of each of the courses is summarized in Tables 1, 2, and 3, respectively. Detailed discussion of the rationale and the process of developing those courses can be found in another published article [6].

The content of each course is organized in a modular fashion, as modules and submodules, which are grouped into a new course, or can be plugged into existing course for the purpose of augmentation. Each submodule may contain one or more instructional units (either lecture or lab). A central repository is created to accommodate all implemented courseware units that are annotated and labeled. This not only helps organize the efforts of applying for the designation in the future but also makes the created content searchable and discoverable. As an introductory course, the Cyber Attacks and Defense course covers a wide range of topics but at rather shallow depth. For instance, many of the network-related topics will be revisited with much more details down the course path. In Tables 1 and 2, we list some sample instructional units to demonstrate the content of the other two new courses, Network Defense and Network Forensics. Both courses are organized in the module → Submodule → Instructional Units structure.

The respective prerequisites of each of the new courses are shown in Fig. 1, which also illustrates how the new courses are integrated into the existent CS curriculum. Hands-on labs were integrated into each of the courses. In addition to utilizing existing labs such as those from the SEED project [2], we developed three sets of labs for the Network Forensics course, including data acquisition labs, data analysis labs, and attack analysis labs.

Table 1 Course design of *Cyber Attacks and Defense*

<i>Module 1. Security fundamentals</i>
<ul style="list-style-type: none"> • Submodule 1: Security concepts and principles • Submodule 2: Security management • Submodule 3: The cybersecurity profession and careers
<i>Module 2. Security threats and countermeasures</i>
<ul style="list-style-type: none"> • Submodule 1: Security threats • Submodule 2: Cyber crimes • Submodule 3: Countermeasures • Submodule 4: Safeguard the IT infrastructure • Submodule 5: Introduction to cryptography
<i>Module 3. Network security</i>
<ul style="list-style-type: none"> • Submodule 1: Networking basics • Submodule 2: Network protocols • Submodule 3: Network administration basics • Submodule 4: Network security basics
<i>Module 4. Software security</i>
<ul style="list-style-type: none"> • Submodule 1: Software vulnerabilities and security • Submodule 2: Low-level attacks and defense • Submodule 3: Secure programming • Submodule 4: Web-based system security
<i>Module 5. Cloud security</i>
<ul style="list-style-type: none"> • Submodule 1: Cloud computing fundamentals • Submodule 2: Cloud security basics

Table 2 Instructional units of *Network Defense*

<i>Submodule: Network defense mechanisms</i>
<ul style="list-style-type: none"> • Network access control • DMZs/proxy servers • Implementing firewalls and VPNs • Application-layer security: HTTPS • Network-layer security: IPSec
<i>Submodule: Network defense hands-on labs</i>
<ul style="list-style-type: none"> • Network sniffing using Wireshark • Implementing IPSec • Setting up honeypots • Securing a web server

3 Method

3.1 Participants

Table 4 displays the student demographics per CDT course. Overall, the majority of students enrolled in the three new CDT courses were Caucasian males in their senior year of a computer science undergraduate program.

Table 3 Instructional units of *Network Forensics*

<i>Submodule: Network technique and forensics</i>	
•	Proxies and forensics
•	Firewalls and forensics
•	NIDS & NIPS and forensics
•	VPN and forensics
•	Router and forensics
<i>Submodule: Network forensics hands-on labs</i>	
•	Tcpdumping with the libpcap library
•	Sniffing wireless traffic with Wireshark
•	Packet sniffing and analysis with NetworkMiner
•	Malware identifying with YARA
•	Evidence acquisition with SNORT
•	Collect and analyze log file with Splunk

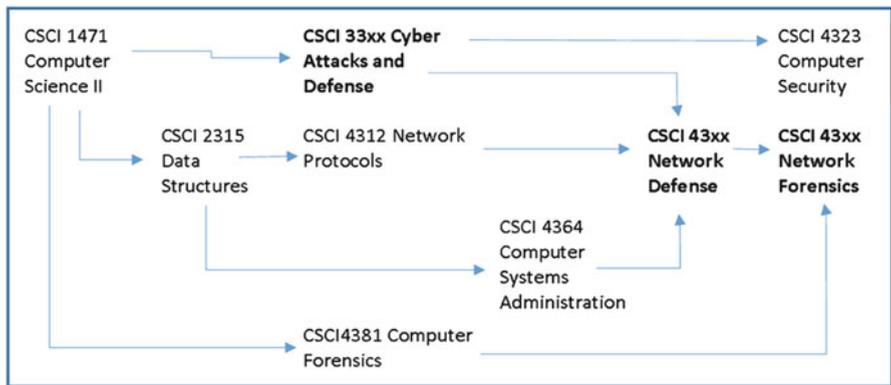


Fig. 1 Integrating the three new courses into the CS curriculum

3.2 Instrumentation

For each of the CDT courses, a researcher-constructed survey was developed and validated by an expert in cybersecurity and computer science to measure cybersecurity awareness, interest in cybersecurity coursework/careers, and attitudes toward active learning. All 27 items were measured via a 4-point Likert scale (*Strongly Disagree; Strongly Agree*). To assess student knowledge and application of each of the three new CDT courses, a researcher-constructed 20-item multiple-choice pre-/post-assessment, aligned to the objectives of the instructed curriculum, was developed for each course. Each assessment was validated by an expert in cybersecurity and computer science prior to administration.

Table 4 Student demographics

	Cyber attacks and defense ^a		Network defense		Network forensics	
	<i>n</i>	%	<i>n</i>	%	<i>n</i>	%
<i>1. Gender</i>						
• Male	9	81.8	10	90.9	8	88.9
Female	2	18.2	1	9.1	1	11.1
<i>2. Race/ethnicity</i>						
Asian	0	0.0	1	9.1	2	22.2
Caucasian	5	50.0	5	45.5	6	66.7
Hispanic or Latino	3	30.0	1	9.1	1	11.1
Indian	1	10.0	0	0.0	0	0.0
Two or more races	1	10.0	3	27.2	0	0.0
Other	0	0.0	1	9.1	0	0.0
<i>3. Grade level</i>						
Sophomore	0	0.0	1	9.1	0	0.0
Junior	2	18.2	2	18.2	0	0.0
Senior	9	81.8	8	72.7	8	88.9
Fifth year	0	0.0	0	0.0	1	11.1
<i>4. College major</i>						
Computer Eng.	1	9.1	1	9.1	0	0.0
Computer information systems (CIS)	2	18.2	2	18.2	1	11.1
Computer science	7	63.6	7	63.6	8	88.9
Information technology	1	9.1	1	9.1	0	0.0

^aOne respondent chose not to provide his/her race/ethnicity

3.3 Data Collection and Analysis

Quantitative and qualitative methods were utilized to collect data using surveys, assessments, and focus groups. Prior to exposure to the CDT curriculum, all students completed a pre-/post-survey to assess interest in cybersecurity-related coursework/careers, cybersecurity awareness, and attitudes toward the use of hands-on learning. A pre-assessment was also administered at the beginning and again at the end of each new course to measure student knowledge and application of cybersecurity. Comparisons between pre- and post-data pointed to the nature of changes in perceptions and knowledge. At the completion of each CDT course, focus groups were conducted to assess student perceptions. All focus groups consisted of 6–9 students and lasted 30–45 minutes. Quantitative data were analyzed using descriptive (frequencies, percentages, averages) and inferential (paired t-tests)

statistics, while qualitative data were analyzed using a constant comparative method. To increase the validity of the results, we triangulated data across all data sources, along with using member checking and peer debriefing.

4 Results

4.1 Knowledge of Cybersecurity

Cyber attacks and defense The results of the two-tailed paired *t*-test indicated that there was not a statistically significant mean difference between the pre- and post-test scores ($t(8) = -1.417, p = 0.194$). Although a statistically significant mean difference was not found, student test scores, however, did increase from before taking the course ($M = 68.5\%$) to the end of the course ($M = 75.0\%$), indicating students did leave the course with increased knowledge (9.5% increase).

Network defense The results of the two-tailed paired *t*-test indicated that there was a statistically significant mean difference between the pre- and post-test scores ($t(6) = -3.732, p = 0.01, d = 1.17$ (large effect), $r^2 = 0.254$). Average student test scores increased from before taking the course ($M = 50.7\%$) to the end of the course ($M = 70.0\%$), indicating students left the course with increased knowledge (38.1% increase). The Network Defense course had a large effect on the students' knowledge, and 25.4% of the variance in the students' post-test scores can be attributed to the course.

Network forensics The results of the two-tailed paired *t*-test indicated that there was a statistically significant mean difference between the pre- and post-test scores ($t(8) = -12.618, p < 0.001, d = 2.83$ (large effect), $r^2 = 0.666$). Average student test scores increased from before taking the course ($M = 50.0\%$) to the end of the course ($M = 80.6\%$), indicating students did leave the course with increased knowledge of network forensics (61.2% increase). The Network Forensics course had a large effect on the students' knowledge, and 66.6% of the variation in test scores can be attributed to the course.

Table 5 presents the results of the paired *t*-tests. The student comments gathered from all of the focus group sessions confirmed these findings. All of the students agreed that the course increased their knowledge in cybersecurity, even those students who admitted to having some foundational knowledge prior to taking the course. One student elaborated further by stating, "In Operating Systems, we just went over the security of the operating system. In Operating Systems, we just touched on the names of SQL injection and the risk factors. In this class, we got a greater understanding of what it actually was. First course we only got the terms and this one we got the definitions, explanations, and examples".

Table 5 Paired *t*-test results per CDT course

CDT course	<i>N</i>	<i>M</i>	SD	<i>t</i> -value	<i>df</i>	<i>p</i> -value	<i>d</i> -value
1. Cyber attacks and defense							
Pre-scores	9	68.5	11.1	-1.417	8	0.194	-
Post-scores	9	75.0	14.1				
2. Network defense							
Pre-scores	7	50.7	19.9	-3.732	6	0.01 ^a	1.17
Post-scores	7	70.0	12.2				
3. Network forensics							
Pre-scores	9	50.0	10.0	-12.618	8	<0.001 ^a	2.83
Post-scores	9	80.6	11.6				

^aStatistically significant (*p* < 0.05)

4.2 Awareness of Cybersecurity Practices

Cyber attacks and defense By the end of the semester, 100% of the students felt they could explain cybersecurity-related key concepts and principles, were aware of the common practice in secure programming, understood the fundamentals of networking, were knowledgeable about the ethical issues in cybersecurity, and could illustrate how privacy is tied to cybersecurity. The largest percent difference from the beginning of the semester to the end was in the students’ abilities to generate a list of security countermeasures. By the end of the semester, 91.0% of the students felt they had the ability to generate a list of security countermeasures in comparison to 45.4% at the start of the semester (100.4% increase), and 36.0% more of the students completed the course believing they were capable of using tools to enhance network security, were aware of cybersecurity-related laws and regulations, and understood cybersecurity in an enterprise setting. When the focus group students were prompted as to whether this course increased their level of cybersecurity awareness, all of them nodded their heads and said aloud, “Yes.” Students commented on how there were no cybersecurity assignments or training aspects to the other courses they had completed. The cybersecurity topics were discussed in those courses, but the students did not actually “do” them. One student elaborated further by stating, “This class provided us with the ‘know how’ of cybersecurity not just the concepts of cybersecurity. Other classes tell you what to do, but not cover the real ‘how to’ and this class did just that.”

Network defense By the end of the semester, 100% of the students felt they could explain key security principles, were aware of the common practice in securing a computer system, and understood the fundamentals of the ISO/OSI 7-layer network model, 72.7% more of the students felt they could generate a list of security countermeasures against attacks at networks, and 63.6% more of the students felt they could enumerate various types of cyber-attacks against a networked system. The largest percent difference from the beginning of the semester to the end was in explaining how the IPSec works. At the start of the semester, 0% of the students

felt they had the ability to explain how the IPSec works in comparison to 90.9% at the end of the semester. When the focus group students were prompted as to whether this course increased their level of cybersecurity awareness, all of them agreed it had. One student elaborated further by stating, "I currently work in web development and I was always kind of interested in cyber-security stuff, but this is the first class I've taken in cyber-security and it definitely increased my interest. I do want to do more and it gave me a positive outlook on the work that can be done in this industry."

Network forensics By the end of the semester, 100% of the students felt they could explain how forensics works and explain key forensics principles, were aware of the common practice in investigating a cyber incident, and understood the difference between network defense and network forensics. 66.7% more of the students completed the course believing they were capable of understanding the analysis algorithms used for network forensics and capable of using tools for network forensics to investigate cyber incidents, while 56.5% more of the students completed the course feeling capable of generating a list of security countermeasures against attacks toward a networked system. The largest percent difference from the beginning of the semester to the end was in the students' abilities to produce a network forensics report. By the end of the semester, 100.0% of the students felt they had the ability to produce a network forensics report in comparison to 0.0% at the start of the semester. When the focus group students were prompted as to whether this course increased their level of cybersecurity awareness, all of them responded, "Yes!" One student commented on how this course increased his awareness of cybersecurity by expanding on previous knowledge, "Before taking the course, I had some knowledge of cybersecurity and the need for it after talking to people in the field both locally and online. This course helped to reinforce how crucial it is to be very disciplined and active."

4.3 Interest in a Cybersecurity Career

Cyber attacks and defense Prior to taking this course, only 54.6% of the students reported having fundamental knowledge of what cybersecurity entailed as a profession, 63.6% felt they were applying security best practices in all of their computing activities, and 63.7% claimed their future career goal was cybersecurity related. By the end of the semester, 100% of the students reported having fundamental knowledge of what cybersecurity entailed as a profession (83.2% increase), 45.4% realized they were applying security best practices in all of their computing activities (28.6% decrease), and 72.8% claimed their future career goal was cybersecurity related (14.3% increase). When asked whether the course increased their interest to pursue a career in cybersecurity, several of the students commented that it had. One student replied, "Initially I was curious, but this course has definitely increased my interest in pursuing it as a career." Another student commented, "My increased

knowledge on the level of escalating and evolving threats and vulnerabilities to computers and networks helped me understand that the job market will always be there - job security.” One student even took his cybersecurity career interest a step further, “I joined the Air National Guard and picked a job in cybersecurity. This class helped me make that choice.”

All of the students were very much in agreement as to the importance and real-world relevance of the hands-on labs. The survey responses shifted from pre to post to 18.2% agreed and 81.8% strongly agreed (18.2% increase) suggesting students’ attitudes toward the importance of the hands-on activities had increased by the end of the semester. Students even commented on the appeal of taking the course based on it included hands-on experience, “The main appeal of taking this class was because of the labs because I had covered the concepts in other courses. I wanted the hands-on experience of actually ‘doing’ the concepts.” Once again, the students reinforced the idea of how this course gave them the “how to” of cybersecurity, “Although we are not tested on the labs, that’s the part you wanted to learn because by ‘doing’ it you will not forget it – it will stick in your mind.”

Network defense Prior to taking this course, 9.1% of the students reported only being interested in the operation aspect of network security, 54.6% felt they were applying security best practices in all of their computing activities, 36.4% claimed they wished they had majored in cybersecurity, and 72.7% are interested in pursuing cybersecurity and/or network security certifications in order to prove to future employers their readiness for real-world challenges in protecting computing systems. By the end of the semester, 36.4% of the students reported only being interested in the operation aspect of network security (300% increase), 81.8% realized they were applying security best practices in all of their computing activities (49.8% increase), 63.6% claimed they wished they had majored in cybersecurity (74.7% increase), and 81.8% are interested in pursuing cybersecurity and/or network security certifications in order to prove to future employers their readiness for real-world challenges in protecting computing systems (12.5% increase).

When asked whether the course increased their interest to pursue a career in cybersecurity, several of the students commented that it did. One of the students commented, “I would rather spend my time doing cyber-stuff than writing programs and developing software.” Another student elaborated further by stating, “I am a computer science major and this course made me realize I have other job options besides being a programmer. I would much rather play with command prompts and set-up firewalls than write code. When looking for a job, I will definitely prioritize looking for an IT job over a programming job.” Students even commented on the impact the labs had on their interest and engagement in cybersecurity, “My interest in the class has come from having to do the hands-on labs and having to figure it out. Seeing it in ‘action’ has built my appreciation. The final product ‘peaked’ my interest; you can see where it works, what it’s actually doing, and what it’s accomplishing.” One student went as far as to comment on the impact the labs had on his career choice, “Doing the labs made me appreciate the IT department more

and what they do. It makes me not terrified of being in IT because it's not just a turn it off then turn it on, unplug then re-plug sort of thing. There is so much more to it."

Network forensics Prior to taking this course, only 33.3% of the students reported having fundamental knowledge of what network forensics entailed as a profession, 44.4% wished they had majored in cybersecurity, and 55.6% were interested in pursuing cybersecurity and/or network security certifications or an advanced degree in order to prove to future employers their readiness for real-world challenges in protecting computing systems. By the end of the semester, 100% of the students reported having fundamental knowledge of what network forensics entailed as a profession (200.3% increase), 66.7% wished they had majored in cybersecurity (50.2% increase), and 77.8% were interested in pursuing cybersecurity and/or network security certifications or an advanced degree in order to prove to future employers their readiness for real-world challenges in protecting computing systems (39.9% increase).

When asked whether the course increased their interest to pursue a career in cybersecurity, all of the students nodded their heads and said, "Yes! For sure." One student replied, "The cybersecurity courses have exposed us to the types of attacks out there and what is going on in the news reports. In the future, it appears there will be a heightened awareness of cybersecurity, especially in the U.S. There will be plenty of jobs." Both the pre- and post-survey responses indicated that 100% agreed that the hands-on activities in cybersecurity-related courses were very important. All of the focus group students were in agreement as to the importance and benefit of participating in the labs. Students commented on the usefulness of the labs to the field, "I found the lab experiences to be particularly useful to the field of cybersecurity. I now know that a lot of the tools used to gather data such as Snort you can just download for free yourself."

5 Discussion

The cybersecurity hiring crisis is directly attributed to the shortage of effective training and education programs, especially in areas that require cyber operations. Traditional computer science undergraduate programs, if augmented with well-designed cybersecurity curricular components, can be a potential remedy to this epidemic problem. In this 3-year-long exploratory project, with the necessary academic rigor, we designed, implemented, and experimented with network security-focused content in our CS curriculum. Assessments discussed in the paper reveal some interesting findings that can be very informative to other institutions with similar agenda and needs:

1. Upper-level traditional CS students may have some cybersecurity knowledge already, but often the knowledge is vague, incomplete, and unsystematic.

2. Explicitly organizing and developing cybersecurity content into meaningful pedagogical tools greatly help students acquire the desired knowledge and skills.
3. Promoting awareness of the profession is as equally important as teaching domain knowledge and skills. This practice helps build the pipeline of future cybersecurity workforce.
4. CS students, with existing computing background, deeply appreciate the hands-on nature of the content. Seeing security in action is exciting and inspirational.

6 Conclusion

In this chapter, we discuss the development and assessment of the three new courses that we developed, in order to create a Cyber Defense Track for our Computer Science undergraduate bachelor program. To promote easy adoption of the new courseware, each course is composed of modules, submodules, and hands-on labs. To assess the effectiveness of the newly developed courseware, we collected data on all the three courses when they were delivered. Both quantitative and qualitative methods were utilized to collect data using surveys, assessments, and focus groups. A survey was developed and administered to measure the students' cybersecurity awareness, interest in cybersecurity careers, and attitudes toward active learning. The survey was administered both before and after the course, in order to compare the students' responses before and after having participated at each of the courses. Comparisons between pre- and post-data pointed to the nature of changes in perceptions and knowledge. As discussed, the assessment results indicated an increased level of (a) cybersecurity knowledge, (b) awareness of cybersecurity-related practices, and (c) interest in a cybersecurity-related career field.

References

1. S. Donovan, B. Cobert, M. Daniel, T. Scott, Strengthening the Federal Cybersecurity Workforce. 31 Oct 2016. Available: <https://obamawhitehouse.archives.gov/blog/2016/07/12/strengthening-federal-cybersecurity-workforce>
2. W. Du, SEED: Hands-on lab exercises for computer security education. *IEEE Secur. Priv.* **9**(5), 70–73 (2011)
3. ISC2, Global cybersecurity workforce shortage to reach 1.8 million as threats loom larger and states rise higher. Available: <https://www.isc2.org/News-and-Events/Press-Room/Posts/2017/06/07/2017-06-07-Workforce-Shortage>
4. M. Mickos, The cybersecurity skills gap won't be solved in a classroom, *Forbes Innovation*. 19 June 2019. Available: <https://www.forbes.com/sites/martenmickos/2019/06/19/the-cybersecurity-skills-gap-wont-be-solved-in-a-classroom/#3a3c15721c30>
5. T. Temin, Alan Paller: Federal progress in cybersecurity. 31 Oct 2016., Available: <http://federalnewsradio.com/federal-drive/2016/06/alan-paller-federal-progress-in-cybersecurity/>
6. W. Wei T.A. Yang, S. Davari, K. Sha, J. Jacob. Toward CAE-CDE 4Y designation through curriculum modernization of a traditional computer science undergraduate program. Proc. of the ISCAP (Information Systems and Computing Academic Professionals). Oct 2018

Team-Based Online Multidisciplinary Education on Big Data + High-Performance Computing + Atmospheric Sciences



Jianwu Wang, Matthias K. Gobbert, Zhibo Zhang, and Aryya Gangopadhyay

1 Introduction

Next to theory and experimentation, computation has become the third pillar [1] and data-driven science has become the fourth pillar of the scientific discovery process [2] for many disciplines and critical to their research advances, such as bioinformatics, physics, computational chemistry, and mechanical engineering. It demands requirements on a course explaining how data and computation related techniques can help scientific discovery. Yet such a “Data + Computing + X” course is often missing in current curriculum design.

In 2017, the National Science Foundation (NSF) published the solicitation “Training-based Workforce Development for Advanced Cyberinfrastructure (CyberTraining)” designed to address this national need. This program continues currently with solicitation number NSF 19-524. The four authors of this paper from three departments across two academic colleges at UMBC joined in response and proposed the UMBC CyberTraining initiative to create the nationwide online team-based training program “Big Data + HPC + Atmospheric Sciences” (cybertraining.umbc.edu) for students in three disciplines (Computing, Mathematics, and Physics) to foster multidisciplinary research and education using

J. Wang · A. Gangopadhyay

Department of Information Systems, University of Maryland, Baltimore, MD, USA

e-mail: jianwu@umbc.edu; gangopad@umbc.edu

M. K. Gobbert (✉)

Department of Mathematics and Statistics, University of Maryland, Baltimore, MD, USA

e-mail: gobbert@umbc.edu

Z. Zhang

Department of Physics, University of Maryland, Baltimore, MD, USA

e-mail: zhibo.zhang@umbc.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_4

advanced cyberinfrastructure (CI) resources and techniques. The course teaches participants how to apply knowledge and skills of high-performance computing (HPC) and big data to solve challenges in Atmospheric Sciences. We focus on the application area of atmospheric physics and within it radiative transfer in clouds and global climate modeling, since these topics are important, pose computational challenges, and offer opportunities for big data techniques to demonstrate their impacts. The NSF funded our proposal in the inaugural year 2017 (OAC-1730250) for training programs conducted in 2018, 2019, and 2020.

Our program is now in its third year, and this paper reports on our experiences in conducting such training online and team-based with participants ranging from undergraduates (NSF-funded through an REU Supplement in Year 3), graduate students, post-docs/non-TT faculty, and TT (tenure-track) junior faculty. We specifically describe how to practically create the necessary training material, chiefly the tapings of lectures for later asynchronous online delivery of contents and homework, during Year 1, and how to accomplish this in an institutionally supportive environment, but without the type of resources an institution with an institutional focus on online teaching would have. Thus, we wish to share our experiences to regular faculty, who might want to add aspects of online teaching to their repertoire. We believe that this is extremely timely information in 2020, where many of us were forced into online teaching with next-to-no notice and no training because of the COVID-19 pandemic.

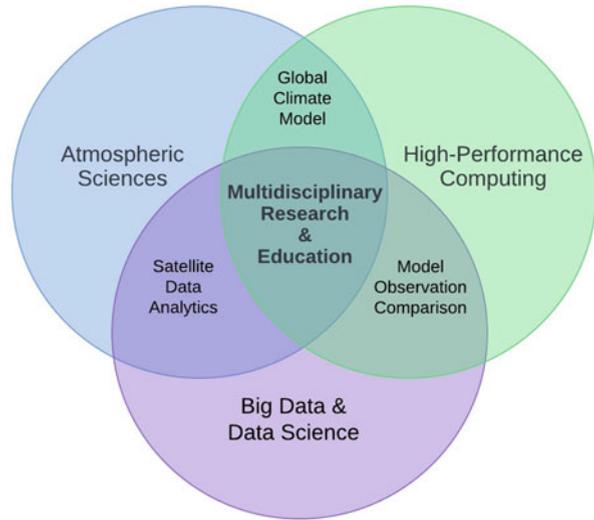
The rest of the paper is organized as follows. In Sect. 2, we explain how our graduate-level course on “Big Data + HPC + Atmospheric Sciences” was designed. Section 3 discusses how we recruited participants from applicants all over the USA. Section 4 focuses on the creation of our online teaching program. Section 5 discusses some challenges we faced and our solutions to them. The paper concludes in Sect. 6.

2 The Big Data + HPC + Atmospheric Sciences Course

As illustrated in Fig. 1, we believe there are a lot connections between big data, HPC, and atmospheric sciences in terms of both education and research topics. So we designed our “Big Data + HPC + Atmospheric Sciences” course through the following innovative approaches: (1) it teaches students in atmospheric sciences how to implement and run parallel and big data programs at an HPC facility; (2) it teaches students in computing and applied mathematics how to solve atmospheric sciences challenges by applying their knowledge; (3) it provides distinctive learning outputs and homework to fit the background and interests of students in different disciplines; (4) it provides team-based frontier research projects where each team is composed with students in different disciplines so they can collaborate and contribute from their own research interests.

Our 15-module multidisciplinary course includes (1) customized course design for three disciplines with commonalities and differences; (2) data and computing techniques adoption for atmospheric sciences (three/four modules each for Data

Fig. 1 Illustration of the connections between big data, HPC, and atmospheric sciences



Science, HPC and Atmospheric Sciences); (3) identification of open challenges (including related open data) that can benefit from advanced CI resources and techniques; (4) 5 weeks long team-based project for frontier research challenges; (5) open source CI software implementation; (6) publications from the designed research projects. During a regular semester, the workload is equivalent to that of a three-credit course and we offer it to UMBC students as a cross-departmental special topic graduate course. The computing environment for the lectures and research projects is provided by our local HPC Facility at UMBC (hpcf.umbc.edu).

Table 1 lists the 15 modules of the course and it takes around 3 h to teach each module. Details of each module are explained below.

Module 1: Introduction of Python/C, Linux, and HPC Environment The first module explains the whole structure of the program and required basic knowledge for the program. It briefly goes through a programming language such as Python or C. It also introduces the hardware architecture, available software, and basic usage of the UMBC HPCF environment (hpcf.umbc.edu).

Module 2: Numerical Methods for Partial Differential Equations (PDEs) This module explains the basics of partial differential equations, which is commonly used in physical models. It discusses the use of numerical methods for PDEs, which is one major driving force behind research in many other fields like numerical linear algebra, scientific computing, and the development of parallel computers. It covers the three basic PDE categories and their mathematical properties with examples. It discusses two large classes of methods: finite difference and finite element methods.

Module 3: Message Passing Interface (MPI) This module explains how to write MPI programs which is one of most common approach to build portable and scalable parallel scientific applications. It covers basic MPI commands such as

Table 1 Modularized structure of our training program

Module	Topic	Goal
1	Introduction of Python/C, Linux, and HPC environment	Running their own jobs on HPC
2	Numerical methods for partial differential equations (PDE)	Model as PDE and solve them using numerical methods
3	Message Passing Interface (MPI)	Write MPI jobs and performance studies
4	Basics of earth-atmosphere radiative energy balance and global warming	Understand basic concepts and principles of radiative energy balance and global warming
5	Basics of radiative transfer simulation framework	Understand the basic physics underlying the transport of radiation in atmosphere
6	Global climate model (GCM) simulation and satellite observations	Understand the importance of GCM and satellite remote sensing
7	Introduction of big data	Understand the basics of big data and demo programs
8	Big data system: Hadoop/Spark	Write Hadoop/Spark jobs and run them on HPC
9	Big data machine learning	Write a machine learning program using Spark MLlib
10	Deep learning	Write a deep learning program
11	Project introduction	20 min project explanation from each team, including Q&A
12–14	Project progress report from each team and feedback	20 min report from each team including Q&A + rating
15	Final project presentation	Technical report, software and a final 30 min presentation from each team (by all team members) including Q&A

MPI_Send and MPI_Recv, collective communication commands like MPI_Bcast, MPI_Reduce/MPI_Allreduce, and MPI_Gather/MPI_Scatter. It also explains how to write MPI programs in both C and Python (through mpi4py).

Module 4: Basics of Earth-Atmosphere Radiative Energy Balance and Global Warming This module explains the basic concepts and principles that control the radiative energy balance of earth-atmosphere system, and its implications to climate. The module starts with the fundamental physics, such as black-body radiation, followed by zero-order radiative energy balance between incoming solar radiation and outgoing terrestrial longwave radiation. The module ends with discussion of what kinds of roles the greenhouse gases, aerosols, and clouds play in the radiative energy budget.

Module 5: Basics of Radiative Transfer Simulation Framework Following the previous module, this module introduces the fundamental physical principles that control the transport of radiation (i.e., visible and infrared light) in our atmosphere. The module also includes the introduction of Monte Carlo method and its application to radiative transfer.

Module 6: Global Climate Model (GCM) Simulation and Satellite Observations This module starts with an introduction to the basic concepts and principles of numerical climate simulations, followed by explaining the importance of evaluating climate simulations and why satellite remote sensing products are invaluable for climate model evaluation. Basic concepts and principles underlying satellite remote sensing are also introduced in this module.

Module 7: Introduction of Big Data This module explains the basic concepts of Data Science, including generic lifecycle and different stages of data analytics, such as acquisition, cleaning, preprocessing, integration, aggregation, analysis, modeling, and interpretation. It explains the basics of big data, including its 5 V characteristics. It starts with the challenges and bottleneck of many applications when dealing with large volume of data. It also covers unique features and challenges for climate/atmospheric data.

Module 8: Big Data Systems: Hadoop and Spark This module covers how to use two popular big data systems, namely Hadoop and Spark. It explains how Hadoop Distributed File System (HDFS) can achieve data partitioning, and fault tolerance and cluster management and job scheduling in Hadoop/Spark. For Spark, it explains resilient distributed datasets (RDD), RDD transformations (map, join, cogroup, etc.) and actions (count, collection, foreach, etc.), lazy evaluation.

Module 9: Big Data Machine Learning This module explains how to conduct machine learning tasks in the above module in a scalable approach through Spark MLlib. Main techniques/concepts include DataFrame-based MLlib API vs. RDD-based MLlib API, ML pipelines, Transformer, Estimator, and Parameter.

Module 10: Deep Learning This module covers deep learning using TensorFlow and Keras. It covers the basics of deep learning such as the network structure, activation functions, optimization, and backpropagation. Specific deep learning models such as convolutional neural networks, recurrent neural networks, and long short-term memory (LSTM) are covered with examples.

Module 11: Project Introduction Each team presents the basics of the research project they will work on in the following 5 weeks. It covers the background, required techniques, suggested phases and major tasks, expected outputs, output evaluation metrics and challenges to each discipline.

Modules 12–14: Project Progress Report from Each Team and Feedback from Instructors These three modules are weekly project progress updates and discussions. Since most teams have three members, every member will be a presenter for the reports. All instructors and other teams discuss the progress, perform peer-review, provide feedback, and give ratings.

Module 15: Final Project Presentation The final module is the final project presentation and final CI software program and technical report delivery. Each team gives a talk on the problems to be solved, the approaches taken, demonstration of developed software program, the experiments and results, and contributions of

each member. All instructors and other teams provide feedback and give ratings and suggestions for future work.

3 Recruitment, Applicants, and Participants

Recruiting for the program used most effectively mailing lists in all three areas that the disciplines are housed in, with a flyer attached. That flyer was also distributed at relevant conferences if faculty or our graduate students attended. The flyers pointed to the program webpage cybertraining.umbc.edu that had program information as well as the link for application. For the three program years, we received 18, 94, and 100 applications, respectively, for the 15 funded participant slots. The recruiting was local and thus the number of applicants was limited in Year 1, since we conducted the training face-to-face, so participants had to be able to travel to UMBC every Friday afternoon.

The participants were selected competitively to form multidisciplinary teams of three participants with generally one participant from each area. The admission of participants was based on demographic information collected in a web form, a CV, a thorough personal statement, and for students with at least two letters of recommendation; upon acceptance of a student, we collected an explicit support from the student's advisor to ensure that the student was allowed to commit time to the program. The personal statement was asked to address specifically why the participant is interested in interdisciplinary research, how participation will promote his/her career goals, and how he/she can contribute to a team of participants from each discipline. The main selection criteria were: (1) how much the applicant can benefit from the training program; (2) how much the application's background is aligned with the program; (3) balanceness among home institutes of applicants. We gave preferences to applicants from underrepresented communities including historically black colleges and universities (HBCUs) and applicants from institutes that have no major HPC facilities. We also strived to make the teams demographically diverse, e.g., with respect to gender of the participants, but kept each team at a relatively consistent educational level. This means that we grouped graduate students of similar class standing together in a team as well as grouped the post-docs/faculty together. This avoids undue difference in leadership experience between the members and also allows to tailor the research project for each team slightly to an appropriate level for each team. This turned out to be a critically useful decision particularly in 2020, when the workload of many post-docs/faculty changed dramatically as their home institutions moved suddenly to online instruction, all while they had small children at home in many cases. It was noticeable that this change affected the graduate students relatively less negatively than the post-docs and faculty (including us faculty on this program ourselves).

The material is at the level of an advanced graduate course, and most participants were graduate students, but as permitted by this NSF program some can also be post-doctoral researchers or junior faculty. For all three groups, participating can

Table 2 Profile of participants for our training program

	Under-graduates	Graduates	Post-docs and non-TT faculty	TT faculty	Total participants	Female participants	Total teams
Year 1	0	9	4	3	16	7	5
Year 2	0	14	2	1	17	6	5
Year 3	6	11	4	4	25	14	8
Total	6	34	10	8	58	27	18

have significant impact on their career in vastly expanding horizons from their own disciplines to two others. After an initial face-to-face course in Year 1 to develop the material, as explained in the next section in more detail, the training in the following years is completely online with participants working together remotely from anywhere in the nation. In this way, this training is available to participants who do not have local access to this kind of material. Another purpose of the face-to-face training in Year 1 was to create a pool of former participants, some of whom could be recruited to work as graduate assistants in Years 2 and 3.

Table 2 summarizes the basic profile of the participants for our program over the three years. We can see (1) most participants are graduate students since we believe graduate students are still in their early years of their research career and the offering of multidisciplinary education would have bigger impacts on their future career growth; (2) we try to address the under-representation of female researchers in STEM disciplines by having relatively equal number of female participants (27) and male participants (31). Some additional participants not eligible for NSF funding (not graduate students or post-docs/faculty) were included without support. An additional benefit for local participants was the three-credit special-topics graduate course.

4 Creation of the Online Training

The fundamental goals of the proposed training were (1) the combination of teams with participants from three disciplines together and (2) to conduct this training online with participants from around the nation. The multidisciplinary nature of the work naturally gives rise to the use of team-based pedagogy.

But how to implement the training online leaves some choices. For instance, a fundamental decision to take is if online training should be completely asynchronous, or if only each team would work synchronously on their own time. We felt that this approach would deprive the teams from experiencing a whole-cohort feeling and we also wanted to foster communication skills. Therefore, the online training includes weekly synchronous meetings on Friday afternoons (Eastern time) that are conducted via Webex or Zoom. It is in principle possible to hold lectures online synchronously. However, it is not the most effective use of valuable

synchronous time. Therefore, we use a flipped-classroom educational model [3], in which the contents are delivered via taped lectures that each participant views asynchronously in their own time. This model applies particularly during the first 10 modules, which constitute the instructional portion of the training. During that time, each team then communicates amongst themselves to organize the work on the team-based homework. The state-of-the-art collaborative and communication tools are used throughout, thus providing deep exposure to skills vital in today's job market. This homework is due by the end of Thursday, so that the faculty can score it on Friday morning. In the synchronous Friday afternoon meeting, each team presents their homework solution to the whole cohort. The goal of the presentations of homework is to familiarize each participant with online presentation and the underlying goal of the team-based homework is to have the teammates gel together. This preparation pays off during the research training phase in Modules 11–15, when teammates now know each other, know each other's strengths, have experience with all communication and presentation technology, and can now present research updates effectively every week, culminating in a complete formal talk like at a conference in the final synchronous meeting. Additionally, the finished technical report from each team is published in the publication series of the UMBC High Performance Computing Facility (hpcf.umbc.edu).

The above describes the pedagogical techniques that we use in the fully online trainings in Year 2 and 3 (2019 and 2020). We feel that it is useful to explain in more detail, how we used Year 1 to conduct a transition from traditional (non-flipped) face-to-face teaching to flipped-classroom online training. Particularly, we wish to communicate here that the approach described in the following is a realistic one for busy research-active faculty with little or no long-term support by technological staff, such as is true at many institutions that do not have an explicit online teaching mission.

To accomplish the transition, we conducted the training in Year 1 (spring semester 2018) face-to-face in a team-taught three-credit course held on campus. This is realistic for workload of the instructors and to give enough time for coordination (preparation during fall 2017 and during the semester itself) among the instructors, who had not taught together before. These synchronous class meetings were traditional lectures and were taped, and these tapes form the basis of the online asynchronous contents' delivery in the following years. These tapings in 2018 were done in an instructional classroom with a camera and initial support from AV staff that set up the equipment. One of the graduate assistants funded by the NSF grant operated the camera, so that the instructor could focus on a normal lecture delivery. We observe that by now (2020), other tools are more widespread, specifically in-cloud software such as Panopto or Blackboard Collaborate that can facilitate taping of lecture delivery within the instructor's own laptop, also during live lecture, thus an assistant would not necessarily be needed any more at all.

All work is conducted in a multidisciplinary team with participants from each area. In the first 10 modules consisting of instruction in all three areas, team building is achieved by homework. In the final 5 modules, each team applies the material learned immediately to a small research project, culminating in a technical

report and a project presentation, by the end of the 15-module program. During the research phase, each team is mentored by a faculty member and supported by a graduate research assistant (RA), who is often a Ph.D. student of the faculty member. During the instructional phase of the first 10 modules, the instructor for each topical area is supported by a teaching assistant (TA) from that area. Many of the TAs and RAs are the same graduate students, but they do not need to be; in particular, some research projects were supported by additional graduate students not paid through this grant. In a similar way, some research projects were collaborations with other researchers, some of whom provided minor and some major leadership and mentorship. In this way, a program like this can be very exciting and invigorating for the research enterprise and even jumpstart new collaborations.

The description so far makes a separation of instruction and research, but we learned the lesson from Years 1 to 2 that it is in fact beneficial to overlap them in time. This means that the research mentor of each team is assigned and known to each team from the start. This mentor reaches out to the team already in the first week, thus giving the team an additional contact person throughout. The mentor can then learn if the team has already some available time and, for instance, start the research by making readings available during the first 10 weeks. In most cases, each mentor holds a weekly separate research project meeting with each team he mentors to discuss research topic, agenda and methodologies. This serves to get a head-start on the research phase and hit the ground running in Week 11. We used this approach in Year 3 to great success in that several teams have already first results in Week 12 that they could report in the synchronous meeting and get feedback from all other participants, earlier than in past years.

5 Discussion

Is Flipped Classroom a Good Teaching Method for Online Instruction? We used face-to-face teaching in Year 1 and flipped online teaching in Years 2 and 3. Comparing their differences, we believe flipped classroom is suitable for online instruction. Our lecturing was most one-direction from instructor to students and the interaction between instructor and students was limited because students were busy following the lecture. By flipping classroom, students study the lecture videos ahead of time and practice their skills via homework before synchronous online discussion. An online Discussion Forum was available for questions throughout and actively monitored by the faculty and TA for the topical area. During the synchronous online meetings, each team presented their homework results and got feedback from instructors and other teams. By doing so, we had more interaction at our synchronous online discussions. Further, each team gets chances to learn how other teams solved the homework problems and compare differences among teams, which we could not do in Year 1's face-to-face teaching because most time was taken for lecturing and no time could be given for in-class homework discussion and presentation.

How to Keep Students Engaged in Online Instruction? One challenge we often face for online instruction is student/participant engagement. Students might feel isolated because they do not see other students like regular face-to-face instruction. It is particularly challenging to this program since most participants are all over the country and each participant might be the only person from his/her institution involved in the program. Also most participants do not know anyone else before the program starts. We addressed the challenge by providing and facilitating various types of communications. The four main communication mechanisms we used are: (1) synchronous weekly online discussion where each team presents their homework or project progress and interact with other teams and instructors; (2) asynchronous web forum discussion for problems raised by students during studying lecture and working on homework and project so questions can answered be as soon as possible; (3) regular online meetings between an instructor and each team he mentors to discuss their progress and problems faces; (4) regular communications within each team to know each other better and collaborate on homework and project. One team reported they had three regular online meetings each week to discuss homework and research. Another team had over 3000 messages via Slack instant messaging (IM) software during the 15-week training period. The third team used WebEx Teams, since that software can hold meetings and save the chat across the whole duration of the program. Overall, we believe communication is the key to keep students engaged throughout the program.

Is It Possible to Complete a Solid Research Project Within Five Weeks? We admit 5 weeks is quite short for a solid research project. In actual implementation of our program/course, the instructors already have identified possible projects before the whole course started so that each team can pick from them if they cannot come up their own project quickly. Further, we encourage teams to discuss and define the project they plan to do early on. In most cases, each team mentor started to have regular weekly research project meetings with each team he mentors in around week 7. We chose this time point for two reasons: (1) team members have been collaborating with each other for a while via several modules and homework; (2) they all have some knowledge of atmospheric science by studying modules 4–6 to understand what could be an application challenge they can work on. We also did not pose hard deadline for the completion of the research project since each project is unique. Even the program/course technically finished after week 15, all teams were willing to continue working on their projects for a few weeks after in order to have good final technical reports. Many teams went on collaborating further to extend their reports to conference/journal papers.

How to Involve Undergraduate Students in a Program Designed for Advanced Graduate Students? In Year 3, we were successful in applying for REU Supplement support for six undergraduate students at our institution from the NSF. We recruited for these positions in August 2019 and admitted two students from each discipline in September 2019. We report on how it is possible to successfully integrate undergraduate students in a program that was conceived for advanced graduate students and junior faculty. The key was to start the training for these local

students during the fall 2019 semester. Since the students had a full course load to start with, the spreading out of material is crucial. The students were grouped by department during fall 2019 with a faculty mentor from that home department. They started by learning about the topics out of the 10 instructional modules that are in their own area, thus when they joined a multidisciplinary team, they had all something to contribute. We then during winter 2019–2020 leveraged the fact that the lecture videos of the first 10 modules are available for asynchronous delivery. The two teams of undergraduates in fact started on the homework and were able to get a head-start of several weeks of homework submissions before the official start of the program. Using the time thus freed up during several weeks of instructions in Weeks 1–10, the undergraduate teams also started on research substantially earlier than Week 11. This concept is currently working, and the undergraduate teams have results on the same level as the more senior teams.

6 Conclusions

Both the National Strategic Computing Initiative [4] and the Federal Big Data Research and Development Strategic Plan [5] highlight the importance of workforce development on HPC and big data. In this paper, we present our efforts of creating a training program or graduate-level online course in big data applied to atmospheric sciences as application area and using HPC as indispensable tool. We outline a concrete procedure how to create the course and believe that this approach could also be used to create other courses for the “Computational and Data Science for All” educational ecosystem. This ongoing program already produced 10 technical reports, 10 peer-reviewed papers [6–14] and a M.S. thesis [15], and most of them are led by participants. The anonymous feedback from participants were also overwhelmingly positive. It reflects, to some extent, the success of our program in its offering of learning and research opportunity to the participants. Also, our experiences on online instruction would be particularly valuable to many instructors during the COVID-19 pandemic.

Acknowledgments This work is supported in part by the U.S. National Science Foundation under the CyberTraining (OAC–1730250) and MRI (OAC–1726023) programs. The hardware used in the computational studies is part of the UMBC High Performance Computing Facility (HPCF).

References

1. Computational Science: Ensuring America’s Competitiveness (2005). https://www.nitrd.gov/pitac/reports/20050609_computational/computational.pdf
2. National Academies of Sciences, E., Medicine, et al., *Future Directions for NSF Advanced Computing Infrastructure to Support US Science and Engineering in 2017–2020* (National Academies Press, Washington, 2016)

3. L. Abeyssekera, P. Dawson, Motivation and cognitive load in the flipped classroom: definition, rationale and a call for research. *Higher Educ. Res. Develop.* **34**(1), 1–14 (2015)
4. The federal big data research and development strategic plan (2016). <https://www.nitrd.gov/Publications/PublicationDetail.aspx?pubid=63>
5. Executive Order – Creating a National Strategic Computing Initiative (2015). <https://www.whitehouse.gov/the-press-office/2015/07/29/executive-order-creating-national-strategic-computing-initiative>
6. P. Guo, C. Liu, Y. Tang, J. Wang, Parallel gradient boosting based granger causality learning, in *2019 IEEE International Conference on Big Data (Big Data)* (IEEE, Piscataway, 2019), pp. 2845–2854
7. C. Barajas, P. Guo, L. Mukherjee, S. Hoban, J. Wang, D. Jin, A. Gangopadhyay, M.K. Gobbert, Benchmarking parallel k-means cloud type clustering from satellite data, in *International Symposium on Benchmarking, Measuring and Optimization*(Springer, Berlin, 2018), pp. 248–260
8. C.A. Barajas, M.K. Gobbert, J. Wang, Performance benchmarking of data augmentation and deep learning for tornado prediction, in *2019 IEEE International Conference on Big Data (Big Data)* (IEEE, Piscataway, 2019), pp. 3607–3615
9. P. Shi, Q. Song, J. Patwardhan, Z. Zhang, J. Wang, A. Gangopadhyay, A hybrid algorithm for mineral dust detection using satellite data, in *2019 15th International Conference on eScience (eScience)* (IEEE, Piscataway, 2019), pp. 39–46
10. H. Song, J. Wang, J. Tian, J. Huang, Z. Zhang, Spatio-temporal climate data causality analytics-an analysis of ENSO’s global impacts, in *Proceedings of the 8th International Workshop on Climate Informatics (CI2018)* (2018)
11. H. Song, J. Tian, J. Huang, P. Guo, Z. Zhang, J. Wang, Hybrid causality analysis of ENSO’s global impacts on climate variables based on data-driven analytics and climate model simulation. *Front. Earth Sci.* **7**, 233 (2019)
12. W. Zhang, J. Wang, D. Jin, L. Oreopoulos, Z. Zhang, A deterministic self-organizing map approach and its application on satellite data based cloud type classification, in *2018 IEEE International Conference on Big Data (Big Data)* (IEEE, Piscataway, 2018), pp. 2027–2034
13. J. Wang, M.K. Gobbert, Z. Zhang, A. Gangopadhyay, G.G. Page, Multidisciplinary education on big data+ HPC+ atmospheric sciences, in *Workshop on Education for High-Performance Computing (EduHPC-17)* (2017)
14. Z. Zhang, H. Song, P.L. Ma, V. Larson, M. Wang, X. Dong, J. Wang, Subgrid variations of the cloud water and droplet number concentration over tropical ocean: Satellite observations and implications for warm rain simulation in climate models. *Atmos. Chem. Phys.* **19**(PNNL-SA-136226), 1077–1096 (2019)
15. C.A. Barajas, An approach to tuning hyperparameters in parallel: A performance study using climate data. Master’s Thesis, Department of Mathematics and Statistics, University of Maryland, Baltimore County, 2019

Integrating the Development of Professional Skills Throughout an ICT Curriculum Improves a Graduate's Competency



Nicole Herbert, David Herbert, Erik Wapstra, Kristy de Salas, and Tina Acuña

1 Introduction

Previous research indicates that graduates from information and communication technology (ICT) courses (degrees) are generally strong in their technical ability but weaker in their ability to communicate and collaborate effectively in the workplace [1–4]. Numerous studies have reported on different approaches for developing professional skills (such as communication and collaboration skills) within an ICT curriculum, and some have shown using examples of one or two subjects where ICT students have developed or improved their professional skills [5–13]. A few studies have reported on a whole of curriculum approach for a single professional skill [1, 2, 14–16]. However, there is no comprehensive analysis of an ICT curriculum-wide approach that evaluates the overall impact on ICT graduates' competency. Evidence that a curriculum-wide approach improves a graduate's competency with professional skills without having a detrimental impact on their level of competency with technical skills could reassure curriculum designers that the approach can lead to professionally competent graduates suitable for employment in the ICT industry.

This chapter reports on a longitudinal study that evaluated the academic outcomes of student cohorts from 2012 to 2018 for a case study ICT curriculum to explore the research question: *How does integrating the development of professional skills across an ICT curriculum impact on a graduate's competency?* This case study ICT curriculum incorporates the core knowledge and skills from the discipline fields of Computer Science (CS), Information Technology (IT) and Information Systems (IS) [17], and the findings of this study are relevant to each discipline.

N. Herbert (✉) · D. Herbert · E. Wapstra · K. de Salas · T. Acuña
University of Tasmania, Hobart, TAS, Australia
e-mail: Nicole.Herbert@utas.edu.au

2 Related Work

The term “competence” links education with work-readiness. Competency is a series of abilities that when combined make a competent person within a professional context. A professionally competent ICT graduate can perform their duties within the ICT profession to the required standard [17]. The ICT graduate has acquired the ability to combine disciplinary knowledge with a range of skills and dispositions to effectively accomplish professional tasks using the employability skill set. The employability skill set is a combination of ICT technical, non-technical and professional skills, suitable for a range of career outcomes [3].

Professional skills (sometimes also called generic, graduate, or soft skills) are transferrable skills that can be applied in different industries and generally take longer to acquire as they require more practice across different ICT domains (such as networking, project management, software development) than technical ICT skills [18]. Professional skills are considered an essential component of employability by employers [19, 20] and include skills like problem-solving, initiative, critical thinking, creativity, digital literacy, communication and teamwork [19, 21, 22].

Graduate employment outcomes can be improved by including professional skill development in the curriculum. One of the key findings from an investigation into whether ICT graduates are ‘work-ready’ (productive) was that most employers consider professional skills to be untrainable in the work environment and consider them a critical hurdle for employment [18, 20, 23]. Palmer et al. [24] confirmed that even though there is significant job growth in the Australian ICT industry, a third of ICT graduates do not find employment in the industry. This implies ICT courses need to equip graduates for productive work in other industries and professional skills are commonly required in most professions [24].

Including the development of the full employability skill set within a curriculum can be difficult. Within ICT there is a tension between balancing discipline-specific content and professional skill development within a standard timeframe of 3 years [25]. Employers are not interested in extending the duration of study for students to acquire professional skills, nor are they interested in solutions that lower a graduate’s technical foundation to acquire professional skills [3, 18].

Many researchers provide examples of how professional skills can be developed at the subject level; however, most do not expand their approaches beyond one or two subjects, or do not attempt to measure the improvement in competency, or do not identify the potential of a curriculum-wide approach. Both Pollock [13] and Hoffman et al. [11] identified how written communication skills could be integrated with technical content without sacrificing the technical skills. Many studies [5, 7, 8, 10, 12] described how they incorporated student presentations in different ICT subjects to develop oral communication skills. Hanna et al. [9] describe how employability skills, including communication and teamwork, were integrated within first-year computing subjects by having employers visibly participate.

A number of researchers have developed approaches for incorporating the development of a single professional skill throughout a curriculum. Falkner and

Falkner [16] present a methodology to integrate written communication skill development with discipline content across the curriculum, and they illustrate its use in a pilot study of two subjects. Their analysis of students that were exposed to the integrated curriculum demonstrated higher overall load pass rates. Coleman and Lang [15] describe a curriculum-wide approach to developing collaboration skills without compromising technical course content. Due to the size of their student body, a quantitative analysis was not possible, though anecdotal evidence and qualitative student feedback lead them to conclude that students learnt to collaborate. Burge et al. [2] provide a framework for curriculum-wide integration of communication and collaboration into ICT curriculum. Anderson et al. [1] demonstrate the use of this same framework for written communication skills in a programming subject, and their results suggested improvement in both written communication and the student's understanding of the technical content. Anewalt and Polack [14] implemented a curriculum-wide approach for oral communication skill development; a survey of randomly selected alumni had 82% of respondents indicating they had adequate oral communication skills for their post-graduation environment.

3 The Case Study Integrated Curriculum

Discussions with more than 30 local ICT industry members revealed that they were satisfied with the University of Tasmania (UTAS) ICT graduates' technical skills but concerned by their weaker professional skills [3]. This led to the design of a renewed ICT curriculum at UTAS that integrated the development of professional skills (communication, collaboration, creativity and critical thinking) alongside the development of the technical skills (such as programming, networking, security and databases) and non-technical ICT skills (project management) to create graduates with a strong employability skill set for a range of careers [3].

Skill development has been integrated across the curriculum [26]. The development of a skill is not restricted to a single unit (a unit is a subject); rather, it is spread across a set of units. When a skill is integrated within a unit, knowledge for the skill is taught and practised together with other knowledge and skills to complete domain-specific activities. Integrating skill development across the curriculum allows more opportunities for reflective development; students obtain depth of learning and make connections across ICT domains into professional practice [15, 16].

Herbert et al. [26] provide a methodology to assist ICT curriculum designers to integrate professional skill development across an ICT curriculum, and the methodology's effectiveness was illustrated with this same case study curriculum.

The outputs (such as presentations, reports and software) from the learning activities within the integrated curriculum to develop and assess professional skills are not particularly novel. What is novel is that students commence development of each professional skill from their first semester and the curriculum allows

the students to have many opportunities to practise each professional skill across different domains. They are provided with formative and summative feedback on their performance allowing for continuous improvement [27]. Another novelty is that the technical, non-technical and professional skills are inextricably woven together to complete a learning task, just as they are in professional practice [25]. Each complex task requires a number of different skills. Competence in the professional, non-technical and technical skills is assessed using the outputs of the activity and observed student behaviours by assessors or by using tools such as peer assessment [28].

4 Methodology

This chapter provides an empirical evaluation of the changes to competency levels for professional skills that are in high demand in the ICT industry [2, 4, 16, 23]: communication (written, oral), collaboration (teamwork, leadership), creativity (entrepreneurship, user experience) and critical thinking (problem-solving, analysis, evaluation, decision-making, reflection). Competency is reflected in a person's behaviour and exhibits itself in the quality of outputs.

Collaboration competency assessment is more complex though, as most work is out of sight of the assessor. Most methods to assess collaboration competency focus on the assessment of group outcomes [29]. To further evaluate the level of competency, two observable behaviours of good teamwork ability that are valued by employers were measured [23]:

- Team members doing their share (contributing)
- Identifying own and others' contributions

To evaluate changes in a graduate's level of competency, comparisons of students' behaviour and the quality of individual and team outputs were made between:

- Students who studied the previous (old) curriculum and those who studied the integrated (new) curriculum
- Students in their first and final year of the integrated curriculum

This longitudinal study compares data using the same students from a first-semester unit KIT105 ICT Professional Practices [30] and their final-year capstone experience where the student teams each complete an authentic software development project for an industry client [27]. The majority of students that completed the capstone experience from 2016 to 2018 (using the unit codes KIT301/KIT302) were enrolled in KIT105 between 2014 and 2016. Comparisons will also be made to the results of students from the capstone experience from 2012 to 2014 (then using the unit codes KXX331/KXX332). These students had not experienced the integrated curriculum. These units were all coordinated by the first author of this chapter.

To complete the evaluation, the following data was collated from KIT105 and the capstone experience units:

- Self- and peer assessment data
- Results from assessment items that were assessed by the same academic (first author)
- Results from capstone assessment items that were assessed by the project clients (industry members)

T-tests are used in the analysis to identify if there is a significant difference between the means of two samples [31]. The unpaired t-test is used when the samples have almost equal variances. Welch's t-test is used for samples when a Levene's test for homogeneity of variance confirmed the variances were unequal [31].

5 Results

Table 1 displays the team and student data for each cohort in this study. Also shown is the number of students from each cohort who completed the entire integrated curriculum. Many students withdraw after completing KIT105 and do not go on to complete KIT302. There are also some students in the capstone experience who come into the course with advanced standing or are completing a discontinued ICT course or are doing an ICT major from another course (e.g. science or business). While these students are not included in the analysis of individual results, they cannot be excluded in the analysis of team results.

5.1 Comparison of Results in Capstone Experience

The capstone experience allows students to apply the knowledge and skills they have acquired throughout the curriculum to a team-based 26-week software development

Table 1 Team and student data

KXX331/KXX332	2012	2013	2014
Total students	61	84	54
Total teams	8	14	8
KIT105	2014	2015	2016
Total students	118	153	124
Entire curriculum	55	70	58
KIT301/KIT302	2016	2017	2018
Total students	53	71	83
Total teams	9	12	12
Entire curriculum	34	65	79

Table 2 *T*-test results comparing the capstone experience learning task outputs

	KXX331/332 2012–2014		KIT301/KIT302 2016–2018		<i>t</i>	<i>p</i>	<i>df</i>
	Mean	Std dev	Mean	Std dev			
Risk log	82.8	8.61	83.48	9.58	0.282	61	0.7783
Business case	79.22	6.58	79	6.07	0.124	46	0.9017
Use-case testing	74.03	13.78	80.58	10.87	2.102	61	0.0397
Design report	74.2	10.9	76.1	8.1	0.818	63	0.4162
Manuals	73.4	7.2	80.4	10.8	3.11	59	0.0029
Presentation client	76.8	14.9	85.6	8.8	2.119	21	0.0461
Presentation lecturer	74.1	9.9	74.7	9.1	0.254	59	0.8
Reflection video	78.4	9.3	73.8	8.5	2.094	63	0.04
Client professionalism	85.2	12.5	86.2	13.2	0.301	53	0.765
Software client	78.6	13.1	84.3	9.96	1.825	52	0.074
Software lecturer	79.7	6.85	78.31	9.77	0.65	63	0.517

project [27]. Teams submit a concept report (week 3) and analysis report (week 5) that include a range of project management documents including a business case and a risk log. Teams submit a design report (week 9), which contains technical design documents (e.g. modelling diagrams, use-case scenarios, interface prototypes). Once the software is nearly complete, the teams prepare a comprehensive use-case testing report (week 24). Finally, teams submit a set of manuals (week 26), consisting of a user guide and a technical maintenance manual.

Unpaired *t*-tests shown in Table 2 indicate there is no statistically significant change in the results for the business case, risk log or design report, but there is a significant improvement for the testing report. A Welch's *t*-test, Table 2, also indicated a statistically significant improvement for the manuals.

Near the end of the capstone experience, teams deliver their final presentation (week 25) which is assessed by the clients and the lecturer. Welch's *t*-tests, Table 2, indicate there is a statistically significant improvement in the industry-client assessment and no significant change in the assessment by the lecturer. Teams also prepare a reflection video that demonstrates their software and, with students speaking to the camera, reflects on their experiences (week 25). An unpaired *t*-test, Table 2, comparing the reflection video results indicates a statistically significant decline.

The clients completed a survey-type assessment on the professionalism of their teams (weeks 13 and 26). The questions concern communication styles and preparedness for meetings. An unpaired *t*-test, Table 2, comparing the professionalism results from the industry clients, indicates the change failed to reach significance.

The final comprehensive software system (week 25) is evaluated by the clients and the lecturer. Unpaired *t*-tests, Table 2, comparing the results indicate no change.

A range of self- and peer-assessment tools facilitate teamwork assessment by measuring observable behaviour [27]. One form of peer assessment has each team member distribute \$100 across team members to give a quantitative opinion of how

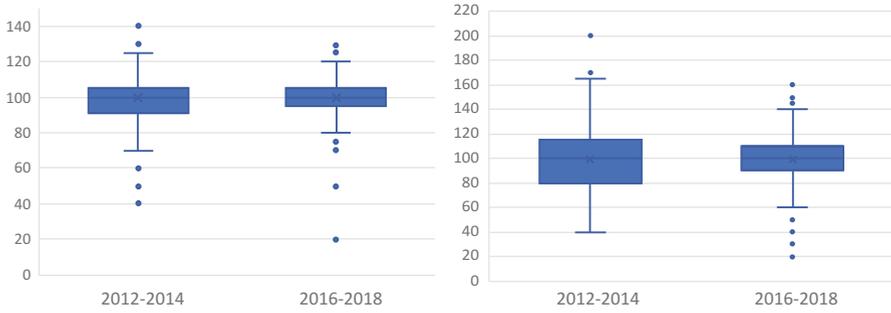


Fig. 1 Design report contribution by peers and software contribution by peers

much each one contributed to an output [27]. A box plot of the amounts for each student for the design report (Fig. 1(a)) and software (Fig. 1(b)) illustrates that under the previous curriculum, the range of total amounts allocated to each student is larger as indicated by the upper and lower quartiles and the whiskers in the plots, signifying that the students are more equally contributing and doing closer to their share in the opinion of their peers in the integrated curriculum.

In 2018, 95% of the students completed the entire integrated curriculum. In 2018, when the majority of the team agreed that a person should get the highest amount, the actual person agreed 92% of the time. When the majority of the team agreed a person should get the lowest amount, the actual person agreed 94% of the time. 74% of the time, the majority of the team gave the same, or more than the person gave themselves. This indicates students can identify their own and others’ contributions.

5.2 Comparison of Results in the First Year and Final Year

KIT105 ICT Professional Practices is a first-year unit that focuses on ethics, law, privacy and entrepreneurship and lays the groundwork for developing graduates who are articulate team members [30]. In KIT105, students prepare an individual job application and a team report on user-centred design. In KIT302, students prepare an individual post-mortem reflection report on their experience (week 26). These documents were not required in the previous curriculum.

A Welch’s t-test, Table 3, comparing the results for the KIT302 post-mortem report for students who had completed KIT105 with those from the first individual document (job application) in KIT105 who went on to complete KIT302 indicates there is a statistically significant improvement in their individual written work. An unpaired t-test, Table 3, comparing the first team document (user-centred design) results in KIT105 with the last team document (the manuals) results in KIT302 indicates there is a statistically significant improvement in their team written work.

Table 3 *T*-test results comparing the first-year and final-year learning task outputs

	KIT105 2014–2016		KIT301/KIT302 2016–2018				
	Mean	Std dev	Mean	Std dev	<i>t</i>	<i>df</i>	<i>p</i>
Individual document	68.5	11.9	74.7	8	5.729	320	<0.0001
Team document	74.8	12.5	80.6	11.6	5.38	577	<0.0001
Presentation	65.5	15.9	74.7	9.1	5.14	56	<0.0001
Video	61.5	20.5	80.6	11.6	13.69	528	<0.0001
Critical thinking report	76.07	12.62	83.5	9.58	3.865	54	0.0003

In KIT105, teams (different team) deliver a presentation about a security breach reported in the media and, in a different team, prepare a speak-to-camera video that pitches an ICT project idea to change the world. A Welch's *t*-test, Table 3, comparing the KIT105 presentation results with the KIT302 presentation results indicates a statistically significant improvement. A Welch's *t*-test, Table 3, comparing the KIT105 video results with the KIT302 reflection video results indicates a statistically significant improvement.

In KIT105, teams prepare an ethical decision report based on a recent ethical event reported in the media (different teams from other activities). In KIT301, the teams prepare an extensive risk log for their capstone project that involves identifying comprehensive contingency plans. Both these documents involve significant critical thinking. A Welch's *t*-test, Table 3, comparing the results indicates a statistically significant improvement.

6 Discussion

It is difficult to perform a controlled longitudinal study, even within one institution. Controlling as much as possible, this study has provided significant evidence that students' competency with professional skills has improved under the integrated curriculum without a decline in their technical skill competency.

6.1 Communication

Written communication skills were integrated, across 14 core units at all year levels within the integrated curriculum [26]. The analysis has shown that students' written communication skills have significantly improved between the first year and final year while experiencing the multitude of opportunities to develop and practice written communication. The results for manuals, where the students have no specific prior practice in the curriculum, have improved after the introduction of writing to learn activities in the form of reports, essays and templates in the first year [26].

Also improved is the software testing report after the above introductions at the first year and the introduction of use case testing into the second year to provide more practice. For both documents, the assessment criteria are unchanged. This confirms the conclusions of [1, 6, 9, 11, 13, 16], with quantitative evidence, that integrating written communication skills curriculum-wide can improve a graduate's competency.

Oral communication skills were only integrated across six core units at all year levels [26]. However, there is evidence that the graduate's oral communication competency has improved. This confirms the conclusions of [5, 8–10, 12, 14], with quantitative evidence, and extends them to a curriculum-wide approach. There was significant improvement between the videos and presentations delivered in the students' first and final semesters, after the students practiced oral communication every year. There were significant improvements between the final presentations assessed by the industry clients for the capstone projects in the previous and integrated curriculum, indicating that the industry is perceiving an improvement. The lecturer evaluation of the presentations showed no change in results. In the integrated curriculum, the lecturer assessment criteria for the presentation have changed to reflect the higher standard that is now required of a graduate, given that they have been delivering presentations from the first year. As the results have not decreased and when considered with the increased evaluations by the industry clients, this indicates that a graduate's competency has improved.

While the change in client professionalism (communication) failed to reach statistical significance, the industry clients have for a long time lauded the professionalism of the teams [27], and so achieving a statistically significant improvement will be a challenge – in 2018, five teams received 100% from their clients; the only other time this was achieved was for one team in 2016.

6.2 Collaboration

Collaboration skills were integrated across nine core units within the integrated curriculum [26]. There is evidence that the graduate's competency with collaboration has improved, confirming the conclusions of [9, 15] with quantitative evidence.

Collaboration competency was first analysed by reviewing the quality of team products. As already discussed, there have been improvements in the team manuals and testing report assessed by the lecturer and team presentations assessed by the industry clients. Collaboration competency was also analysed by reviewing two behaviours, both of which indicated competency within the cohort.

6.3 Creativity

Creativity skills (development was not included in previous curriculum) were integrated across four core units within the new curriculum [26]. Creativity skill development involved introducing entrepreneurship and user-centred design throughout each year of the curriculum. Evidence of improvement was indicated by the comparisons between KIT105 and KIT302 creativity items: team document, presentation and video. The majority of items in the capstone experience that use creativity skills indicate either improvement or no evidence of decline: design report (includes prototypes of interfaces), manuals (user guide), presentation and software. The anomaly, already discussed, is the decline for the reflection video.

6.4 Critical Thinking

While development of critical thinking skills was featured highly in the previous curriculum, these were strengthened in the early years of the integrated curriculum [26]. Mathematics was introduced at the first year to improve fundamental analysis and evaluation skills. Hands-on network laboratory exercises were included at the second year to develop practical problem-solving skills. Ethics was moved to the first year to introduce decision-making early into the curriculum. Evidence of improvement was indicated by the comparisons between KIT105 and KIT302 critical thinking items: individual and team document and ethical-decision report versus risk log. The items in the capstone experience that use critical thinking skills indicate either improvement or no evidence of decline: business case, risk log, design report, testing report, manuals and software.

6.5 Technical Skills

A detrimental impact on technical skill competence is a concern of many academics and industry members when integrating professional skill development alongside the development of technical skills [3, 13, 16, 18, 25]. While this chapter only includes the results for some documents, t-tests were calculated for each technical document in the capstone experience where knowledge and skills are acquired at the second year via an in-house project and then practised again during the authentic project in a professional context. These technical aspects of software analysis and design were also covered at the second year in the previous curriculum, so all cohorts had prior practice with these technical documents, and the assessment criteria are unchanged. There was no significant decline in the results on any documents, and the software testing report and final manuals improved. This study also found no evidence of a decline in the results for the complex software systems

created by each team. This indicates technical skill competency has remained at acceptable levels achieved by the previous curriculum [3].

7 Conclusion and Future Work

This longitudinal study has addressed the research question: How does integrating professional skills throughout an ICT curriculum impact on a graduate's competency? The statistically significant results provide strong evidence that the integration of professional skill development across a curriculum can enhance the employability of graduates by improving competency with professional skills. This study has also shown that there has been no degradation in a graduate's technical ability after integrating professional skill development and retaining a course duration of 3 years.

References

1. P. Anderson, S. Heckman, M. Vouk, D. Wright, M. Carter, J. Burge, G. Gannod, CS/SE instructors can improve student writing without reducing class time devoted to technical content: Experimental results, in *Proceedings of the 37th International Conference on Software Engineering - Volume 2*, vol. 2, (IEEE Press, Piscataway, 2015), pp. 455–464
2. J. Burge, G. Gannod, M. Carter, A. Howard, B. Schultz, M. Vouk, D. Wright, P. Anderson, Developing CS/SE students' communication abilities through a program-wide framework, in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, (ACM, New York, 2014), pp. 579–584
3. N. Herbert, K. de Salas, I. Lewis, J. Dermoudy, L. Ellis, ICT curriculum and course structure: the great balancing act, in *Proceedings of the Sixteenth Australasian Computing Education Conference*, vol. 148, (Australian Computer Society, Inc., 2014), pp. 21–30
4. T. Koppi, F. Naghdy, *Managing Educational Change in the ICT Discipline at the Tertiary Education Level* (Australian Learning and Teaching Council, 2009)
5. C. Bennett, T. Urness, Using daily student presentations to address attitudes and communication skills in CS1, in *Proceedings of the 40th ACM Technical Symposium on Computer Science Education*, (ACM, New York, 2009), pp. 76–80
6. L. Blume, R. Baecker, C. Collins, A. Donohue, A “communication skills for computer scientists” course, in *Proceedings of the 14th annual ACM SIGCSE Conference on Innovation and Technology in Computer Science Education*, (ACM, New York, 2009), pp. 65–69
7. J. Börstler, O. Johansson, The students conference—a tool for the teaching of research, writing, and presentation skills, in *Proceedings of the 6th Annual Conference on the Teaching of Computing and the 3rd Annual Conference on Integrating Technology into Computer Science Education: Changing the Delivery of Computer Science Education*, (ACM, New York, 1998), pp. 28–31
8. I.I. Douglas Dankel, J. Ohlrich, Students teaching students: incorporating presentations into a course, in *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, (ACM, New York, 2007), pp. 96–99
9. P. Hanna, A. Allen, R. Kane, N. Anderson, A. McGowan, M. Collins, M. Hutchison, Building professionalism and employability skills: Embedding employer engagement within first-year computing modules. *Comput. Sci. Educ.* **25**(3), 292–310 (2015)

10. J. Havill, L. Ludwig, Technically speaking: Fostering the communication skills of computer science and mathematics students, in *Proceedings of the 38th SIGCSE Technical Symposium on Computer Science Education*, (ACM, New York, 2007), pp. 185–189
11. M. Hoffman, P. Anderson, M. Gustafsson, Workplace scenarios to integrate communication skills and content: A case study, in *Proceedings of the 45th ACM Technical Symposium on Computer Science Education*, (ACM, New York, 2014), pp. 349–354
12. G. McDonald, M. McDonald, Developing oral communication skills of computer science undergraduates, in *Proceedings of the Twenty-fourth SIGCSE Technical Symposium on Computer Science Education*, (Indianapolis, 1993), pp. 279–282
13. L. Pollock, Integrating an intensive experience with communication skills development into a computer science course, in *Proceedings of the Thirty-second SIGCSE Technical Symposium on Computer Science Education (SIGCSE '01)*, (ACM, New York, 2001), pp. 287–291
14. K. Anewalt, J. Polack. 2017. A curriculum model featuring oral communication instruction and practice. In *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, New York, 33–37
15. B. Coleman, M. Lang, Collaboration across the curriculum: A disciplined approach to developing team skills, in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, (ACM, New York, 2012), pp. 277–282
16. K. Falkner, N. Falkner, Integrating communication skills into the computer science curriculum, in *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education*, (ACM, New York, 2012), pp. 379–384
17. Association for Computing Machinery, Association for Computing Machinery Curricula Recommendations (2019), <http://www.acm.org/education/curricula-recommendations>. Accessed 15 Apr 2019
18. M. Stevens, R. Norman, *Industry Expectations of Soft Skills in IT Graduates: A Regional Survey* (Australasian Computer Science Week Multiconference, 2016)
19. M. Exter, S. Caskurlu, T. Fernandez, Comparing computing professionals' perceptions of importance of skills and knowledge on the job and coverage in undergraduate experiences. *ACM Trans. Comput. Educ.* **18**(4), Article 21 (2018), 29 pages
20. D. Finch, L. Hamilton, R. Baldwin, M. Zehner, An exploratory study of factors affecting undergraduate employability. *Education + Training* **55**(7), 681–704 (2013)
21. J. Coldwell-Neilson, Assumed Digital Literacy Knowledge by Australian Universities: Are students informed? in *Proceedings of the Nineteenth Australasian Computing Education Conference*, (ACM, New York, 2017), pp. 75–80
22. AIIA, Skills for today. Jobs for tomorrow (2017), https://www.aiaa.com.au/_data/assets/pdf_file/0020/81074/JOBS-FOR-TOMORROW-FINAL.pdf
23. M. Hamilton, A. Carbone, C. Gonsalvez, M. Jollands, Breakfast with ICT employers: What do they want to see in our graduates? In *Proceedings of the Seventeenth Australasian Computing Education Conference (ACE2015)*, Sydney, Australia, 27–30 Jan 2015
24. S. Palmer, J. Coldwell-Neilson, M. Campbell, Occupational outcomes for Australian computing/information technology bachelor graduates and implications for the IT bachelor curriculum. *Comput. Sci. Educ.* **28**(3), 280–299 (2018)
25. R. Al-Mahmood, P. Gruba, Approaches to the implementation of generic graduate attributes in Australian ICT undergraduate education. *Comput. Sci. Educ.* **17**(3), 171–185 (2007)
26. N. Herbert, T. Acuna, K. de Salas, E. Wapstra, A methodology to integrate professional skill development throughout an ICT curriculum, in *25th Annual ACM Conference on Innovation and Technology in Computer Science Education Conference*, (ACM, New York)
27. N. Herbert, Reflections on 17 Years of ICT capstone project coordination: Effective strategies for managing clients, teams and assessment, in *Proceedings of the 49th ACM Technical Symposium on Computer Science Education*, 21–24 Feb 2018, Baltimore, MD, USA
28. N. Clark, P. Davies, R. Skeers, Self and peer assessment in software engineering projects, in *Proceedings of the 7th Australasian Conference on Computing Education*, (Newcastle, 2005), pp. 91–100

29. M. Conde, F. Rodríguez-Sedano, L. Sánchez-González, C. Fernández-Llamas, F. Rodríguez-Lera, V. Matellán-Olivera, Evaluation of teamwork competence acquisition by using CTMTC methodology and learning analytics techniques, in *Proceedings of the Fourth International Conference on Technological Ecosystems for Enhancing Multiculturality*, ed. by F. J. García-Peñalvo, (ACM, New York, 2016), pp. 787–794
30. N. Herbert, Impact of student engagement on first year ICT performance, in *2017 International Conference on Computational Science and Computational Intelligence (CSCI)*, (Las Vegas, 2017), pp. 1085–1090
31. Social Science Statistics, Statistics calculators (2019), <https://www.socscistatistics.com/tests/>. Accessed 24 Nov 2019

Preparing Computing Graduates for the Workplace: An Assessment of Relevance of Curricula to Industry



Ioana Chan Mow, Elisapeta Mauai, Vaisualua Okesene, and Ioana Sinclair

1 Introduction

The current Computing curriculum at the National University offers courses in the areas of computer applications, computer programming, information systems, and networking as part of Certificate, Diploma, and Bachelors of Science degree in Computing and Applied Computing as well as a Postgraduate Diploma Science in Applied Computing. The certificate, diploma, and bachelor program follow a stair-casing arrangement with 8 courses for the certificate, 11 courses for the diploma, and 24 courses for the bachelor program. The department also offers service courses in Computing to other faculties within the university. The Computing curriculum is continually undergoing revisions. At the outset of curriculum development, it was decided to offer this combination of computer applications, programming, information systems, and networking courses under a Computing major rather than a Computer Science major as this combination of courses was deemed as more relevant to the needs of the local industry and society [9].

The motivation to undertake this research arose primarily from an interest in the effectiveness and relevance of the programs offered by the Computing Department in order to meet the needs of the local industry and society. In addition, there was a need to evaluate the effectiveness of the changes in the curriculum instituted as a result of the 2008 and 2014 graduate survey [4, 5]. The proposed survey (RINCCIII) builds on these two previous surveys and assesses whether the revisions to the curriculum resulted in achieving “more relevance” to workplace needs. The goal is that the outcomes of this research would provide information about the relevance of the current content within the curricula and can be applied within the context

I. C. Mow (✉) · E. Mauai · V. Okesene · I. Sinclair
Computing Department, National University of Samoa, Apia, Samoa
e-mail: i.chanmow@nus.edu

of improving our course offerings within our programs. The proposed research attempts to answer the following question:

How relevant is the content of Computing courses offered within undergraduate programs of the Computing Department of the National University of Samoa to meet the needs of industry and the workforce?

A major issue and concern is the need to avoid obsolescence and adapt to changing needs within the industry. Information Communication Technology is a field which is not only diverse but also dynamic and rapidly changing [11, 12, 19]. Furthermore, in recent times, the Computing curriculum at NUS has come under some heavy criticism from a 2012 and 2017 PSC survey [14, 15] and consultations for the Communications Sector Plan [13] as not producing quality graduates for the workplace. The need to align our courses with the needs of industry is well articulated within the following documents: (i) The Faculty response report to the Faculty External Review Report 2014, 2020; (ii) Research Report on “An investigative study on the ‘relevance’ of Computing courses at NUS to the needs of the Industry and the workplace, 2015”; (iii) PSC survey 2012, 2017; (iv) Communications Sector Plan 2017–2023; and (v) Faculty of Science Annual Management Plan 2018.

Within the context of the proposed study, relevance is defined in terms of whether the Computing skills and knowledge acquired by students from our courses can be effectively utilized to fulfill their occupational responsibilities. Relevance is evaluated using two measures: (a) frequency of usage of the various technologies and (b) perceptions of relevance of both skills and processes and methodologies expressed as (i) percentage of perception of respondents who found the curricula as relevant or very relevant and (ii) mean of perceptions of relevance with a value range from 1 to 4.

2 Literature Review

Computer science is an enormously vibrant field. From its inception just half a century ago, computing has become the defining technology of our age. In fact the IT industry is now widely applicable in all sectors of the industry [17]. The field, moreover, continues to evolve at an astonishing pace. As new technologies are introduced, existing ones become obsolete almost as soon as they appear [6, 7].

It is frequently suggested that computer science curricula are generated in a vacuum with little or no regard for the “real-world needs” of the student’s ultimate employer. Members of the academic and industrial communities are continually discussing the issue of the “gap” between the training computer science graduates receive in academic institutions and the background industry requires of its new employees in computing-related positions. Such discussions generally fall into one of two categories: (i) descriptions of the gap and (ii) curriculum descriptions.

2.1 Descriptions of the Gap

Discussions are frequently held at educational technology conferences and workshops which attempt to characterize the gap by describing the industry's needs and evaluating computer science curricula with respect to meeting these needs. The resulting descriptions of this gap are usually stated in rather broad terms, such as the following: Computer science graduates cannot solve large, "real-world" computing problems, adequately document their work, or function well as members of a team [6, 7].

2.2 Curriculum Descriptions

Many articles exist in the computing education literature [8, 10] which describe new courses, usually of the work/study or internship variety, and which attempt to bridge the academic/industry gap. The emphasis in such courses is usually not on a particular area of computer science (such as operating systems) but rather on the areas mentioned above, that is, "real-world" problems, documentation, and teamwork. Thus as argued by Govender and Naicker [10], any ICT curriculum renewal strategy must seek to engage with documents that identify ICT areas of need. These areas of need can be triangulated with feedback from industry practitioners. The design and development of new curriculum must cater for the skills required by the twenty-first-century learner. CC2005 [8] categorized twenty-first-century skills internationally as ways of thinking such as critical thinking and problem-solving; ways of working which include communication and collaboration; and tools for working which refer to ICT and information literacy and skills for living in the world such as citizenship and social responsibility.

These views are reiterated in the Computer Science Curriculum 2013 report [7] and Information Technology Curricula 2017 [12] which include examples of ways in which an undergraduate Computer Science program encourages the development of soft skills and personal attributes. These abilities include teamwork, verbal and written communication, time management, problem-solving, and flexibility as well as risk tolerance, collegiality, patience, work ethic, and appreciation for diversity. They all play a critical role in the workplace and in promoting successful professional practice in a variety of career paths. The above views are also supported by studies by Radermacher et al. [16] and Williams [20].

The Computer Science Curriculum 2013 report [7] organized computer science around 18 knowledge areas that reflect the application of computing tools in a wide array of disciplines. It also incorporates new areas of knowledge for computing skills that include information assurance and security, parallel and distributed computing, and platform-based applications. The report provided curricular models suitable to a broad range of higher-education institutions worldwide.

According to the report, Computer science education too often focuses on individual rather than on managed group efforts that depend on defined standards, methodologies, and software processes. However, such group efforts are the norm in the software industry. New graduates often know little about what are regarded as “best practices” in the software engineering profession (e.g., practices related to the use of software processes, measurement and analysis, team building, front-end development methods, quality engineering, software maintenance, and testing) [6, 7].

This problem of inadequate preparation of current graduates is further intensified by the increasing demand for software engineers and other computing professionals [2, 3].

Technical advances over the past decade have increased the importance of many curricular topics, such as the following: (i) The World Wide Web and its applications; (ii) Internet of Things; (iii) networking technologies, particularly those based on TCP/IP; (iv) graphics and multimedia; (v) embedded systems; (vi) relational databases; (vii) interoperability; (viii) object-oriented programming; (ix) the use of sophisticated application programming interfaces (APIs); (x) human-computer interaction; (xi) software safety; (xii) security and cryptography; (xiii) application domains; and (xiv) artificial intelligence.

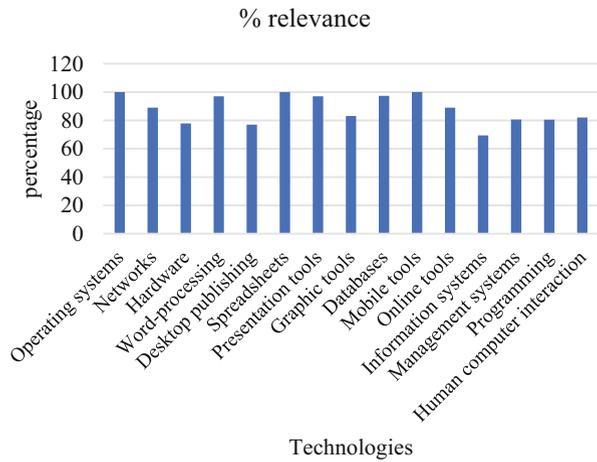
As these topics become increasingly important, it is tempting to include them as undergraduate requirements. Unfortunately, the restrictions of most degree programs make it difficult to add new topics without taking others away. It is often impossible to cover new areas without reducing the amount of time devoted to more traditional topics whose importance has arguably faded with time. The Computing Curricula 2005 Task Force [8] has therefore sought to reduce the required level of coverage in most areas so as to make room for new areas.

Computing education is also affected by changes in the cultural and sociological context in which it occurs. The following changes, for example, have all had an influence on the nature of the educational process: (i) changes in pedagogy enabled by new technologies, (ii) the dramatic growth of computing throughout the world, (iii) the increasing economic influence of computing technology, (iv) greater acceptance of computer science as an academic discipline, and (v) broadening of the discipline. Additionally, in recent times, the effects of climate change, natural disasters, and the COVID-19 pandemic have redefined work and education and magnified even more the importance of technology globally in what is now described as the “new normal” ([21]).

From the scope of the literature, the key players in determining relevance of Computing curricula to the needs of local industry and society are (i) the graduates, (ii) the users of technology, (iii) the industry and ICT professionals, and (iv) the discipline. Relevance of the computing curricula is also influenced by factors such as (v) pedagogy and (vi) technology (refer to Fig. 1). The proposed survey evaluates the perceptions of relevance of these key players and factors and as such decides the target respondent groups for this survey (refer to “Methodology” section).

Firstly, the survey evaluates relevance by gauging the frequency of usage of applications and tools that had been taught during the course of the curriculum.

Fig. 1 Percentage relevance of topics



Secondly, the survey investigates the relevance of courses to the needs of the industry by evaluating the usefulness of skills and knowledge within the program content to the skills and knowledge used in the industry and the workplace. Thirdly, the survey gauges relevance of courses to industry by assessing whether computing processes and methodologies taught in the curricula have been useful and applicable to the industry and workplace.

Hence relevance within the context of this research is defined and measured in terms of (i) frequency of usage of taught applications and tools within the industry and workplace and (ii) usefulness of skills and knowledge, computing processes, and methodologies taught to the skills and knowledge, computing processes, and methodologies used in the workplace. It must be noted that these two levels of measuring relevance overlap and are interrelated.

3 Methodology

The current research is investigative and quantitative in nature. The intention was to attempt to survey all NUS Computing graduates currently working in Samoa and institutions within the CBD. However, due to time constraints and poor response in the early stages, the research team only surveyed those Computing graduates employed in Upolu. In the end, the team was able to obtain 36 responses. Of these, 20 were from NUS Computing graduates working in different institutions in urban Apia and 16 from institutions. Graduates ranged from lecturers to ICT officers, programmers, systems administrators, and some Heads (Assistant CEO) of ICT divisions. Institutions surveyed ranged from various government ministries, colleges, corporations, to banks.

Two versions of the survey were distributed. One version was for institutions, and the second version was distributed to NUS Computing graduates. The only difference between the two versions is the first section on background information and personal information. With the exception of some minor changes, the proposed survey is the same as the one which was administered in 2008 and 2013.

The survey was conducted in Upolu within the CBD as the bulk of ICT usage is within this area. For each institution which employed NUS graduates, surveys were distributed to immediate supervisors of NUS Computing graduates and current students and also to NUS Computing graduates and/or current students. For institutions which utilize ICT but do not employ NUS graduates, surveys were distributed to the head of the ICT section or department. In conducting the survey, consent was sought and participants assured of the confidentiality of the information provided.

4 Analysis

Data from the surveys was mostly empirical data on the frequency of usage and relevance of curricular topics, but there were also open-ended questions on areas in the curricula needing coverage and/or improvement.

In terms of inferential analysis, one-way ANOVA did not reveal any significant differences in terms of frequency of usage as well as perceptions of relevance in terms of age or respondent type (graduate or institution). Hence, there were no significant differences in responses between graduates and institutions. There were also no significant differences in responses by age. Independent samples *t*-testing indicated, however, significant gender differences in terms of frequency of usage of databases with females indicating greater usage than males ($N = 36$, $t = 3.07$, $df = 33$, $p = 0.004$). There were also significant gender differences in perceptions of relevance of desktop publishing applications with males indicating greater perceptions of relevance than females ($N = 36$, $t = -2.669$, $df = 32$, $p = 0.012$).

5 Results and Discussion

5.1 *Systems Present in Institutions*

There was an increase in the number of systems utilized in the various institutions when compared to the 2013 survey and 2008 survey. In the 2008 survey, very few had communication technologies such as videoconferencing. The 2019 survey indicated also more specialized systems such as vehicle tracking, border management systems, GIS systems, and local and regional court databases SAMLI and PACLI.

Such information is important in guiding the skills required and what systems need to be taught within the computing curricula.

5.2 Relevance

As mentioned earlier, two measures were used to evaluate the relevance of the curricula to the workplace: (a) frequency of usage of the various technologies and (b) perceptions of relevance of both skills and processes and methodologies expressed as (i) percentage of perception of respondents who found the curricula as relevant or very relevant and (ii) mean of perceptions of relevance ranging from 1 to 4 (1 = not relevant, 2 = marginally relevant, 3 = relevant, 4 = very relevant).

Frequency of Usage of Technologies

The most frequently used technologies such as database, World Wide Web, email, mobile phone, and word processing were used by most of the institutions on a daily basis, while at the other end of the spectrum, multimedia and programming and graphics were used the least frequently.

The mean frequency of usage across all participants was used as a measure of relevance with the higher means indicating greater usage (refer Table 1). On average, the most frequently used and hence the most relevant topics were World Wide Web, email, mobile phone, word processing, and spreadsheets. The least used and topics with the least relevance were programming and multimedia development. These responses were very similar to the responses from the previous graduate surveys phones [4, 5] However, there was now increasing usage of social networks and mobile phone.

Percentage Relevance

Percentage relevance is defined as percentage of participants with responses of “relevant” or “very relevant.” In terms of percentage perceptions of relevance (refer to Table 2), respondents found the majority of the aspects of the various topics in the curricula either relevant or very relevant to their employment in the workplace with percentage of perceptions of relevance ranging from 69.4% to 100% and mean of perceptions of relevance ranging from 3.08 to 3.69. Using percentage relevance and mean of perceptions of relevance as measures, respondents found operating systems, mobile tools, spreadsheets, databases, word processing, and presentation tools as most relevant but programming, information systems, and human computer interaction relatively not as relevant.

Table 1 Summary of frequency of usage of various technologies

Technology	<i>N</i>	Mean	Std dev
Database applications	36	4	1
World Wide Web	36	3.97	0.17
Email	36	3.92	0.55
Mobile phone	36	3.83	0.69
Word processing	36	3.81	0.4
Spreadsheet	36	3.72	0.45
Social network	36	3.64	0.76
Scanners	36	3.44	0.84
Handheld devices	36	3.42	0.87
Presentation tools	36	3.28	0.74
Digital camera	36	3	0.9
Desktop publishing	36	2.86	0.99
Videoconferencing	36	2.69	0.95
Graphics program	36	2.61	0.84
Multimedia programs	36	2.33	1.12
Computer programming	36	2.22	1.05

Table 2 Summary of perceptions of relevance of technology topics taught

Topic	% relevance	<i>N</i>	Mean	Std dev
Spreadsheets	100	36	3.69	0.5
Word processing	97.2	36	3.67	0.5
Databases	97.3	36	3.64	0.5
Presentation tools	97.2	36	3.61	0.5
Operating systems	100	36	3.56	0.5
Networks	88.9	36	3.5	0.78
Mobile tools	100	18	3.44	0.5
Online tools	88.9	18	3.44	0.7
Graphic tools	83	36	3.4	0.9
Hardware	77.8	18	3.39	0.98
Management systems	80.6	36	3.22	1.1
Desktop publishing	77	36	3.17	0.9
Human computer interaction	82	36	3.15	1.1
Programming	80.5	36	3.11	1
Information systems	69.4	36	3.08	1.1

Recommendations for Improvement

The goal of the current study was to evaluate how relevant the Computing courses taught at NUS are to the needs of the industry and the workplace. The outcomes of this research have provided useful information about the relevance of the current content within the curricula and can be applied within the context of improving our course offerings within our programs. More importantly, the survey identified various aspects which need to be considered for inclusion in the curricula to achieve

better alignment with industry needs. This is the “gap” referred to in the literature [1, 18]. The next few sections outline features of each topic used in the workplace as well as recommendations for improving recommendations for improvement.

Operating Systems

Most of the users have used file management, control panel settings, and installation programs, including remote desktop connections, firewall, disk defragmentation, cleanup, and management services, the setup of network and user accounts, and the use of the device manager. Operating systems used were mostly Windows 7, Windows 10, Windows Server 2012, Windows Server 2016, and Linux. There was also a noticeable increase in the level of complexity of skills mentioned as well as the range of skills and knowledge when compared to previous surveys. In the area of operating systems, suggestions for improvement were extensive and included (i) task management, (ii) networking capability, (iii) hardware adaptability, (iv) file management, (v) hardware interdependence, and (vi) error detection and protection. However, all these have been integrated in the Hardware and Networking courses HCS187, HCS287, and HCS387.

Hardware

Features of hardware used in the workplace include file management, installing new devices and operating systems, repair of PCs and printers, installing software updates, and virus updates. Recommended areas for inclusion in the curriculum included installing updates, troubleshooting hardware errors, and programming routers and switches. These however are areas already covered in the curriculum (HCS187, HCS287, HCS387).

Networking

Features of networking used in the workplace include network design, troubleshooting networks, routing protocols, making and laying cables, programming routers and switches, Internet connections, broadband network bridges, remote access, setting up of IP addresses (IPv4 and IPV6) and IP domains, and permissions. In addition to the basic networking features, there was also mention of connecting to CCTV, wireless routers, VPN, and network virtualization. Feedback on networking indicated the need for knowledge of basic network concepts and a strong need for more practicals on networking skills such as cable connections; LAN, VLAN, and WAN connections; and Linux, emphasis on security and cybersecurity, emphasis on problem-solving skills, as well as the teaching of DOS for troubleshooting.

Word Processing

Word-processing skills used in the workplace of the 36 respondents range from basic formatting to advanced features such as smart art, watermarks, webpages, graphics, styles, macros, and mail merge. For word processing, suggestions to teach up-to-date versions were irrelevant as programs were currently upgraded to Office 2016. There were also suggestions to integrate Word with cloud-based storage.

Desktop Publishing

Features of desktop publishing tools used in the workplace are covered in the section on graphic tools. Desktop publishing is now explicitly taught at the degree level at NUS as part of the Applied Computing major alongside Graphic Design (HCS188, HCS288, HCS388) and also as part of the Certificate of Computer Operating (polytechnic level) which prepares students to be computer operators and data entry operators. Also, web publishing is taught as part of the Human Computer Interaction course (HCS386). Recommendations for improvement in desktop publishing include (i) design and layout of brochures for flyers, pamphlets, wall calendars, and manuals/annual reports through the use of Adobe InDesign, CS5, and Publisher; (ii) web publishing; (iii) teaching of video-editing software, animations, and graphics; and (iv) need for a more practical-based approach to teaching. These recommendations are irrelevant as these have all been incorporated into the new courses on graphic design and animations.

Spreadsheets

Responses indicated extensive use of spreadsheets in the workplace. Features of spreadsheets used in the workplace range from basic skills such as formulae, graphs, tables, and filters to more advanced features such as merging, pivot tables, exporting, importing data, linked worksheets, statistical analysis, CSV format for uploading and creating users for email, and Moodle. All of the NUS graduates indicated that all spreadsheet features covered in their degree are adequate for their work needs.

Presentation Tools

The most widely used presentation tool in the workplace is PowerPoint. Majority of the respondents indicated that all of the features covered as part of the curriculum adequately meet their needs in the workplace. Recommendations for presentation tools and graphic tools were irrelevant as they have all been incorporated into the new courses in graphic design and animation as well as the generic course HCS182.

Graphics

Features of graphics tools used in the workplace included design of logos, brochures, postcards, posters, advertising, business cards, notices, newsletters, invitations, graphics websites, and slide shows, image retouching, and photo editing. Software used include Dreamweaver, Adobe Photoshop and Illustrator, and Publisher. Graphic tools are taught as part of the new Applied Computing major in the Bachelor of Science. Graphic tools are also taught though at the trades level Certificate in computer operations. The majority of the respondents recommended the inclusion of Adobe Photoshop, Macromedia, and Flash which are already incorporated into the new Graphic Design courses HCS188, HCS288, and HCS388.

Databases

Features of databases used in the workplace include creating databases, queries, forms, reports, macros, drag and drop programming, SQL programming, and searching databases. The point to note is relatively to other topic areas, most of the content of databases covered in the curricula are used in the workplace. Aspects not covered in the syllabus at undergraduate level is web integration of databases (SQL server 2016), and while Access is used in the curricula, some workplaces are using other database management systems such as Oracle. In the area of databases, suggestions for improvement included more focus on SQL programming and introduction to CPro, Oracle, MySQL, and other database platforms and web-based database applications.

Online Tools

Online tools include online marketing, online utilities (e.g., backup and data storage), online transaction processing, online surveys, as well as online social forums, e.g., Facebook, email, and chat. There were no recommendations for improvement of coverage of online tools.

Mobile Tools

Mobile tools used in the workplace include video, chat, text, email, scheduler, as well as a variety of mobile apps such as videoconferencing (Messenger, FaceTime, Skype). There were no recommendations for improvement of coverage of mobile tools.

Information Systems

The majority of respondents indicated that they used information systems, and features used were all features currently taught within the curriculum. These included systems development life cycle, case tools, systems analyses, systems design, and systems evaluation. Respondents also gave examples of information systems such as Spiceworks used for IT helpdesk management. Feedback on information systems recommended coverage of different types of information systems as well as a focus on database systems. However, database systems is already the focus of the curriculum in this area.

Management of Information Systems

Features of management of information systems used in the workplace were all features taught in the curriculum: Porter's competitive forces, value chain, Rockarts CSF, Nolan's stages of growth, Mintzberg's theory on bureaucracy, ICT strategic planning, risk management, and disaster recovery. On the subject of management of information systems, recommendation is for a stronger emphasis on security and on strategic planning and risk management as there is a strong push in the ministries for these skills.

Computer Programming

Programming languages used in the workplace included Visual Basic, Java, JavaScript, VBScript, Html, .Net, PHP, SQL, RPG, C++, and Drupal. It must be noted that compared to previous surveys, there has been a noticeable increase in the use of programming languages with all respondents indicating use of one or more programming languages. The single most prominent use of programming languages in the workplace is for programming web pages. Recommendations for programming included (i) inclusion of Perl, Python, PHP, and JavaScript in the curriculum, (ii) programming to be taught at an earlier age, and (iii) more database programming. JavaScript, however, is already taught in the undergraduate courses as part of human computer interaction and web design.

5.3 Human Computer Interaction

Features used in the workplace include interface design life cycle, design of interfaces (forms, reports), evaluation of interfaces, web page design, use of visual basic for programming interfaces, and windows widgets with the majority of the respondents indicating web design as the major use of HCI in the workplace. In the area of human computer interaction, recommendations included the need

to offer web design at NUS which has already been implemented. An important recommendation is the need to cover web-based integration with databases as this aspect is missing from the curriculum.

6 Summary and Conclusion

In summary, the current study has shown that the current Computing curriculum is to a large extent relevant to the needs of industry. When compared to the findings of the earlier surveys, the curriculum was now “more relevant” due to changes made based on recommendations of these earlier surveys. There are recommendations for improvements in the curriculum, identified in this study which need to be taken on board such as emphasis on operating systems, cybersecurity, strategic planning, problem-solving skills, and introduction of other database platforms such as Oracle as well as more web-based languages such as PHP. However, on the overall, most of the recommendations have already been included in the previous surveys and already included in the curriculum. This is a significant improvement from the last two graduate surveys where recommendations resulted in the incorporation of operating systems, hardware and graphics design, and more networking practicals. The recommendation for more emphasis on problem-solving supports the literature and the importance of not just “hard skills” but also “soft skills” [16, 20]. Although teamwork and problem-solving are taught to some extent in project-based work, the emphasis on problem-solving needs urgent attention. As indicated by recent reports from the World Economic Forum on jobs [22], the need for problem-solving skills will become increasingly important over the next few years especially in jobs in the ICT sector. Subsequently there also needs to be inclusion in future surveys of some evaluation on soft skills. Also, one of the recommendations in the IT 2017 Curriculum review [12] is the focus of the curriculum on learning outcomes and competencies. This points to the importance of the graduate profile, which is an area that has not been included in the scope of past surveys and studies. Hence, future surveys also need to include specific consideration of the graduate profile and learning outcomes. The scope of the three surveys has been limited to the undergraduate programs, and it is recommended that future surveys be extended to include an evaluation of the postgraduate program.

The NUS Computing Department have already put in place some of the recommendations from the research literature for improving relevance and closing the gap between the curricula and industry needs. These include the establishment of a CISCO academy to supplement graduate skills in IT; project-based courses to provide teamwork skills and project experience; case studies to provide real-life context and problem-solving; the inclusion of industry representatives on the curriculum advisory board; the introduction of operating systems, hardware, and graphics design into the undergraduate program; the establishment of a center of excellence in IT to provide continuing professional development in areas which are

developing rapidly; and the regular conduct of this survey to gauge relevance of our Computing curriculum to industry.

The main recommendations for improving the curriculum are (i) an emphasis on operating systems, cybersecurity, strategic planning, and problem-solving skills, (ii) introduction of other database platforms such as Oracle, as well as (iii) more web-based languages. Recommendations for improvement of the survey are (i) the increase of scope to include an evaluation of the postgraduate diploma, (ii) the inclusion in future surveys of questions on soft skills, and (iii) specific consideration of the graduate profile and learning outcomes.

References

1. A.N. Ayofe, A.R. Ajetola, Exploration of the gap between computer science curriculum and industrial IT skills requirements, 2009. arXiv preprint arXiv:0908.4353
2. D.J. Bagert, The challenge of curriculum modelling for an emerging discipline: Software engineering education, in *Proceedings of the Frontiers in Education Conference*, (IEEE Computer Press, Tuscan, 1998)
3. D.J. Bagert, Computing education 2020: Balancing diversity with cooperation and consistency, in *Computer Science Education in the 21st Century*, ed. by T. Greening, (Springer Verlag, New York, 2020)
4. I.T. Chan Mow et al., The relevance of computing courses at the National University of Samoa (NUS) to the needs of industry and the workplace. *J. Samoan Stud.* **3**, 2010 (2010)
5. I.T. Chan Mow, H. Sasa, E. Mauai, M. Tanielu, F. Faamau, Relevance to industry needs of computing courses at the National University of Samoa: the RINCCII study. *J. Syst. Cyber. Inf.* **13**(1) (2015)
6. CS 2008, *IEEE/ACM Joint Task Force on Computer Science Undergraduate Curricula 2008* (CS, 2008)
7. CS 2013, *IEEE/ACM Joint Task Force on Computer Science Undergraduate Curricula 2013* (CS, 2014)
8. CC 2005, *IEEE/ACM Joint Task Force on Computing Curricula 2005* (CC, 2005)
9. Faculty of Science, *Response to the External Review Report* (National University of Samoa, 2003)
10. A. Govender, K. Naicker, Designing and developing ICT curriculum in the 21st century using a modernistic curriculum model in contemporary higher education. *Mediterranean Journal of Social Sciences* **5**(23) (2014). MCSER Publishing, Italy
11. IS 2010, *IEEE/ACM Joint Task Force on Computing Curricula. Information Technology 2010, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology* (ACM and IEEE-Computer Society, 2010)
12. IT 2017, *IEEE/ACM Joint Task Force on Computing Curricula. Information Technology 2010, Curriculum Guidelines for Undergraduate Degree Programs in Information Technology* (ACM and IEEE-Computer Society, 2017)
13. MCIT, Communications sector plan 2017/2018–2021/2022 (2017)
14. PSC 2012, PSC Survey report on findings of the survey of government institutions on perceptions of quality of graduates in the workplace (2012)
15. PSC 2017, PSC Survey report on findings of the survey of government institutions on perceptions of quality of graduates in the workplace (2017)
16. A. Radermacher, G. Walia, D. Knudson, Investigating the skill gap between graduating students and industry expectations, in *Companion Proceedings of the 36th International Conference on Software Engineering*, (ACM, 2014), pp. 291–300. <https://doi.org/10.1145/2591062.2591159>

17. G.L. Sipin, J.L.D. Espiritu, O.A. Malabanan, Issues on the Philippines' information and communications technology (ICT) competitiveness (2014), http://www.dlsu.edu.ph/research/centers/aki/_pdf/_concludedProjects/_volumeI/Sipinetal.pdf
18. C.B. Simmons, L. Simmons, Gaps in the computer science curriculum: An exploratory study of industry professionals. *J. Comput. Sci Coll.* **25**(5), 60–65 (2010)
19. A.B. Tucker, Computing curricula 1991: Report of the ACM/IEEE-CS joint curriculum task force, in *Working Group on Software Engineering Education and Training*, (IEEE Computer Society Press, 1996)
20. A.C. Williams, *Soft Skills Perceived by Students and Employers as Relevant Employability Skills* (Walden University, 2015)
21. World Economic Forum, *Report on Challenges and Opportunities in a Post COVID-19 World*, vol 2020 (World Economic Forum, 2020)
22. World Economic Forum, *Executive Summary. The Future of Jobs* (World Economic Forum, 2016), p. 2016

Benchmarking the Software Engineering Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE Software Engineering Body of Knowledge (SWE Knowledge Areas #6–10)



Moh'd A. Radaideh

1 Introduction

The Software Engineering Undergraduate Program (SWE-Curriculum) at Jordan University of Science and Technology (JUST) recently acquired and obtained an accreditation from the Institute of Engineering and Technology (*IET*) [6]. However, the curriculum of the said program needs further expansion to ensure its readiness for any potential ABET accreditation [5] in the future as well as its readiness for training programs, professional licensing, and certification of specialties in Software Engineering. The SWEBOK-V3.0 [3] of the IEEE Computer Society [4] introduced 15 Software Engineering Knowledge Areas (SWE-KAs). Some of them are not fairly covered or addressed in the said SWE-Curriculum. Table 1 lists these 15 SWE-KAs. Table 2 lists the Software Engineering courses (SWE-Courses) of the said SWE-Curriculum [2].

This paper is a continuation of a previous paper (*P#1*) by the author in which the SWE-KAs#1–5 were addressed [1]. As shown in Table 1, this paper represents the second part (*P#2*) of the three parts of this research. It covers the second five (SWE-KAs#6–10) of the fifteen SWE-KAs. The third paper (*P#3*) shall cover the last five (SWE-KAs#11–15) of the fifteen SWE-KAs.

The SWE-KAs that are addressed in this paper are:

This paper is the second of three parts of the benchmarking research that the author has been carrying on.

Md. A. Radaideh (✉)
Jordan University of Science and Technology, Irbid, Jordan
e-mail: maradaideh@just.edu.jo

Table 1 SWE-KAS (Software Engineering Body of Knowledge, SWEBOK-V3.0)

P#1		P#2		P#3	
SWE-KA#	SWE KAs (SWEBOK-V3.0) [3]	SWE-KA#	SWE KAs (SWEBOK-V3.0) [3]	SWE-KA#	SWE KAs (SWEBOK-V3.0) [3]
1	Software requirements	6	Software configuration management	11	SWE professional practice
2	Software design	7	SWE management	12	SWE economics
3	Software construction	8	SWE process	13	Computing foundation
4	Software testing	9	SWE models and methods	14	SWE math. Foundation
5	Software maintenance	10	Software quality	15	Engineering foundation

Table 2 The SWE courses of the SWE-Curriculum at JUST

SWE Courses at JUST (SWE-Curriculum) [1]					
SE210	Java Programming [47]	SE321	Software Requirements Eng [47]	SE430	Software Testing [50]
SE220	Software Modelling [48]	SE323	Software Documentation [48]	SE431	Software Security [51]
SE230	Fund. of Software Engineering II [43]	SE324	Software Architecture & Design [49]	SE432	Software Engineering for Web Applications [52]
SE310	Visual Programming [44]	SE326	Software engineering lab 1 [55]	SE440	Project Management [53]
SE320	Systems Analysis and Design [45]	SE471	Client/Server Programming [56]	CS318	Human-computer interaction (<i>Elective</i>) [54]
SE441	Software Quality Assurance [46]				

1. *SWE-KA#6: Software Configuration Management.* Chap. 6 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [10–12].
2. *SWE-KA#7: Software Engineering Management.* Chap. 7 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [13–19].
3. *SWE-KA#8: Software Engineering Process.* Chap. 8 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [20–24].

4. *SWE-KA#9: Software Engineering Models and Methods*. Chap. 9 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [24, 25].
5. *SWE-KA#10: Software Quality*. Chap. 10 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [26–36].

Section 3 provides the details of our research approach (in the three parts of this research, *P#1*, *P#2*, and *P#3*) involving the reflection of the various topics of these SWE-KAs onto the various courses of the said SWE-Curriculum. It is worth mentioning that although this paper measures the coverage of the said SWE-Curriculum with the SWE-KAs, our innovative approach is general and can be applied to other Software Engineering academic programs.

The findings of this paper (*P#2*) demonstrated a decent degree of compliance in the cases of the *Software Engineering Management SWE-KA* [13–19] and the *Software Quality SWE-KA* [26–36] and a partial compliance in the cases of the *Software Configuration Management SWE-KA* [10–12], the *Software Engineering Process SWE-KA* [20–24], and the *Software Engineering Models and Methods SWE-KA* [24–25]. As the *Software Quality Assurance* is currently an elective course in the said SWE-Curriculum, this paper recommended to make this course a core rather than an elective course in the said SWE-Curriculum.

This paper is organized in several sections. Section 1 is the Introduction. Section 2 discusses the related work. Section 3 elaborates on the research methodology followed to carry on this research work. Section 4 composes five subsections where each of them elaborates on the coverage of one of the second five SWE-KAs in the SWE-Curriculum courses. Section 5 summarizes the findings of this paper, and a set of recommendations is made for possible enhancements on the said SWE-Curriculum to make it more compliant with the second five of the fifteen SWE-KAs and thus to improve its readiness for any potential ABET Accreditation in the future. Section 6 presents the conclusions of this paper.

2 Related Work

Early efforts toward organizing the teaching of Software Engineering include, but not limited to, a paper by Bernhart, M. et al., “Dimensions of Software Engineering Course Design” [7], and another one by Shaw, M. “Software Engineering Education: A Roadmap” [8]. Nevertheless, it is very important that Software Engineers read through the *The Mythical Man-Month* book of Brooks FP [9].

Qiu et al. [37] illustrated the problem-based learning approach that adopts a blended learning environment, a combination of a face-to-face learning environment and an eLearning environment for teaching undergraduate software engineering

principles as well as collaborative skills. They surveyed applying the problem-based learning approach that shows a vast student-felt comfortable learning using problem-based learning, and their educational performances were also better than anticipated.

Garousi et al. [38] investigated challenges facing fresh Software Engineering graduates early in their professional careers. Their study claimed such challenges are due to misalignment of the skills gained through their undergraduate period. They discussed the consequence of Software Engineering graduates not having practice along with soft skills in general before starting their careers, the value of certain SE activities, and abilities in SE education (especially requirements engineering, design, and testing).

Bastarrica et al. [39] surveyed software engineering students regarding relative importance and challenge of different dimensions entailed in their projects. They found out that the comparable value of soft skills develops. At the same time, that of the technical challenge falls and that the surveyed students found planning of the projects and collaboration more troublesome than they anticipated. Also, they found statistically notable evidence that, for the soft skills they have measured, the perceived corresponding relevance changes throughout the course. The surveyed students tend to undervalue the difficulty involved in teamwork before starting their Capstone Course. Therefore, they assumed that their students will be more alert to this concern while handling the upcoming projects.

Barzilay et al. [40] proposed a multidimensional Software Engineering course framework that is organized through four axes: fundamentals of SE, practices and tools, productization, and technology evolution. Their proposed work support students to have a comprehensive cross paradigm and at the same time provides practical and theoretical experience. Each axis enables an examination of Software Engineering from different perspectives. They also describe their experience of teaching the course three times in the Tel Aviv University and the academic college of Tel Aviv-Yafo, Israel.

Dekhane et al. [41] showed how to fill the gap in a project that needs knowledge in two different domains by proposing the integration of these two domains and to have one interdisciplinary project. The authors evaluated their proposed work by providing software engineering students with authentic experiences involved.

Daimi et al. [42] proposed a model that faculty can incorporate in their Software Engineering courses. The innovational approaches of brainstorming, critical thinking, case methods, problem-based learning, trimming techniques, and opportunity recognition will be introduced. Each approach will be supplemented by some examples from the Software Engineering domain. These approaches and their accompanying examples aim to develop several entrepreneurial-mindset attributes.

3 Research Methodology

The research approach followed to carry out this research (in *P#1*, *P#2*, and *P#3*) work can be outlined in the following steps:

1. Dividing the SWE-KAs into the following two groups:
 - (a) *Specialization SWE-KAs* (SWE-KAs#1–10)
 - (b) *Support SWE-KAs* Knowledge Areas (SWE-KAs#10–15).
2. Splitting (*due to the size of this research work*) the *Specialization SWE-KAs* (SWE-KAs#1–10) into two parts such that the first part (*P#1*) covers the first five SWE-KAs (SWE-KAs#1–5) and the second part (*P#2*) covers the second five SWE-KAs (SWE-KAs#6–10). Consequently, the third part (*P#3*) will cover the *Support SWE-KAs* (SWE-KAs#11–15).
3. Inspecting the coverage of the SWE-KAs (*in each of P#1, P#2, and P#3*) in the said SWE-Curriculum.
 - (a) The coverage of the *Specialization SWE-KAs* Knowledge Areas (*P#1* and *P#2*) will be inspected across the SWE Specialization courses across the said SWE-Curriculum.
 - (b) The coverage of the *Support SWE-KAs* Knowledge Areas will be inspected across the University with the College required courses across the said SWE-Curriculum.
 - (c) The syllabus of each course in the said SWE-Curriculum will be carefully reviewed to figure out its coverage of the various topics of the various SWE-KAs.
 - (d) The latest version of the said SWE-Curriculum (*the IET Accredited SWE-Curriculum*) will be used for this research work.
4. Classifying the coverage of each SWE-KA (in *P#1, P#2, and P#3*) in the said SWE-Curriculum into one of the following levels:
 - (a) *Fully Compliant* (100%). This indicates that the concerned SWE-KA is fully covered across one or more of the courses of the said SWE-Curriculum.
 - (b) *Highly Compliant* (75%–<100%). This indicates that the concerned SWE-KA is highly covered across one or more of the courses of the said SWE-Curriculum.
 - (c) *Partially Compliant* (50%–<75%). This indicates that the concerned SWE-KA is partially covered across one or more of the courses of the said SWE-Curriculum.
 - (d) *Poorly Compliant* (<50%). This indicates that the concerned SWE-KA is poorly covered across one or more of the courses of the said SWE-Curriculum.
5. Classifying the coverage of the main topics of each SWE-KA (in *P#1, P#2, and P#3*) in the courses learning outcomes (CLOs) of the said SWE-Curriculum. The learning outcomes are obtained from the syllabi of the courses of the said SWE-

Curriculum, which can be accessed at [1]. The CLO coverage of each SWE-KA is classified into one of the following levels:

- (a) *Fully Compliant* (100%). This indicates that all main topics of the concerned SWE-KA are fully declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
 - (b) *Highly Compliant* (75%–<100%). This indicates that most of the main topics of the concerned SWE-KA are declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
 - (c) *Partially Compliant* (50%–<75%). This indicates that part of the main topics of the concerned SWE-KA are declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
 - (d) *Poorly Compliant* (<50%). This indicates that few of the main topics of the concerned SWE-KA are declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
6. Shortage identification and making recommendations. At the end of each part (*P#1*, *P#2*, and *P#3*), the coverage compliances (or shortages) will be identified, and recommendations will be made such that new courses shall be introduced into the curriculum or existing ones shall be enhanced and/or revised in the said SWE-Curriculum.
 7. Verifying the achievement of the research prime objective. The overall purpose of this work is to facilitate the potential ABET Accreditation of the SWE Undergraduate Program of JUST.

4 SWE-KAS Coverage in the SWE-Curriculum at JUST

The following set of tables (Tables 3, 4, 5, 6, and 7) illustrate the coverage of each of the second five SWE-KAs in the various Software Engineering Courses (SWE Courses) at JUST.

4.1 Coverage of the SWE-KA#6 (Software Configuration Management)

Table 3 concludes the following:

1. The SWE-KA#6 (Software Configuration Management) seems to be *Partially Compliant* in the said SWE-Curriculum through the SE441 Software Quality Assurance course. The SE441 course addresses the various Software Configuration Management topics, but without going into details.

Table 3 SWE-KA#6 (SW Configuration Management) and its coverage in JUST SWE-Curriculum

Software Configuration Management KA (SWEBOK-V3.0)	Partially covered in
1. Management of the SCM process	SE441
Organizational context for SCM	SE441
Constraints and guidance for the SCM process	SE441
2. Planning for SCM	SE441
SCM plan	SE441
Surveillance of software configuration management	SE441
Software configuration identification	SE441
Identifying items to be controlled	SE441
Software library	
3. Software configuration	SE441
Requesting, evaluating, and approving software changes	SE441
Implementing software changes	SE441
Deviations and waivers	SE441
4. Software configuration status accounting	SE441
Software configuration status information	SE441
Software configuration status reporting	SE441
5. Software configuration	SE441
Software functional configuration audit	SE441
Software physical configuration audit	SE441
In-process audits of a software baseline	SE441
6. Software release management and delivery	SE441
Software building	SE441
Software release management auditing	SE441
7. Software configuration management tools	SE441

2. None of the main topics of the SWE-KA#6 (Software Configuration Management) contributes to the learning outcomes of the SE441 Software Quality Assurance course.

4.2 Coverage of the SWE-KA#7 (Software Engineering Management)

Table 4 concludes the following:

1. The SWE-KA#7 (Software Engineering Management) seems to be *Fully Compliant* in the said SWE-Curriculum through the SE440 Software Engineering Management course. The SE40 course covers the various Software Engineering Management topics.

Table 4 SWE-KA#7 (SWE Management) and its coverage in JUST SWE-Curriculum

SWE Management KA (SWEBOK-V3.0)	Fully Covered in	CLOs Relevance
1. Initiation and scope definition	SE 440	CLO1-to-CLO6 of the SE440
1.1 Determination and negotiation of requirements	SE 440	
1.2 Feasibility analysis	SE 440	
1.3 Process for the review and revision of requirements	SE 440	
2. Software project planning	SE 440	
2.1 Process planning	SE 440	
2.2 Determine deliverables	SE 440	
2.3 Effort, schedule, and cost estimation	SE 440	
2.4 Resource allocation	SE 440	
2.5 Risk management	SE 440	
2.6 Quality management	SE 440	
2.7 Plan management	SE 440	
3. Software project enactment	SE 440	
3.1 Implementation of plans	SE 440	
3.2 Software acquisition and supplier contract management	SE 440	
3.3 Implementation of measurement process	SE 440	
3.4 Monitor process	SE 440	
3.5 Control process	SE 440	
3.6 Reporting	SE 440	
4. Review and evaluation	SE 440	
4.1 Determining satisfaction of requirements	SE 440	
4.2 Reviewing and evaluating performance	SE 440	
5. Closure	SE 440	
5.1 Determining closure	SE 440	
5.2 Closure activities	SE 440	
6. Software engineering measurement	SE 440	
6.1 Establish and sustain measurement commitment	SE 440	
6.2 Plan the measurement process	SE 440	
6.3 Perform the measurement process	SE 440	
6.4 Evaluate measurement	SE 440	
6.5 Software engineering management tools	SE 440	
6.6 Matrix of topics vs. reference material	SE 440	

All of the main topics of the SWE-KA#7 (Software Engineering Management) contribute to the learning outcomes of the SE440 Software Engineering Management course.

4.3 Coverage of the SWE-KA#8 (SWE Process)

Table 5 concludes the following:

Table 5 SWE-KA#8 (SWE Process) and its coverage in JUST SWE-Curriculum

SWE Process KA (SWEBOK-V3.0)	Covered in	CLOs Relevance
1. Software process definition	SE441/SE324/SE230	
Software process management	SE230	
Software process infrastructure	SE230	
2. Software life cycles	SE324/SE230	
Categories of software processes	SE230/SE440	CLO4
Software life cycle models		
Software process adaptation	SE230	
Practical considerations	SE430/SE230	
3. Software process assessment and improvement	SE230	CLO4 of the SE440
Software process assessment models	SE230	
Software process assessment methods	SE230/SE440	
Software process improvement models	SE230	
Continuous and staged software process ratings	SE230	
4. Software measurement	SE324/SE230/SE441/SE323	
Software process and Product measurement	SE230	
Quality of measurement results	SE441	
Software information models		
Software process measurement techniques	SE230	
5. Software engineering process tools	SE440	CLO4 of the SE440

1. The SWE-KA#8 (Software Engineering Process) seems to be *Partially Compliant* in the said SWE-Curriculum through the following courses: (i) SE230 Fundamentals of Software Engineering, (ii) SE324 Software Architecture and Design, (iii) SE323 Software Documentation, (iv) SE440 Software Project Management, and (v) SE441 Software Quality Assurance.
2. Some of the topics of the SWE-KA#8 (SWE Process) contribute to the learning outcome CLO4 of the SE440 course, while none of these topics contributes to the learning outcomes of the SE320, SE324, SE323, and SE441 courses.

4.4 Coverage of the SWE-KA#9 (SWE Models and Methods)

Table 6 concludes the following:

- 1- The SWE-KA#9 (SWE Models and Methods) seems to be *Partially Compliant* in the said SWE-Curriculum through the following courses: (i) SE220 Software Modelling, (ii) SE321 Software Engineering Requirements, (iii) SE324 Software Architecture and Design, (iv) SE440 Software Project Management, and (v) SE441 Software Quality Assurance.
- 2- All of the main topics of the SWE-KA#9 (SWE Models and Methods) contribute to the learning outcomes of the SE220 course.
- 3- The last two topics of the SWE-KA#9 (SWE Models and Methods) contribute to the learning outcome CLO2 of the SE440 course.

4.5 Coverage of the SWE-KA#10 (Software Quality)

Table 7 concludes the following:

- 1- The SWE-KA#10 (Software Quality) seems to be *Fully Compliant* in the said SWE-Curriculum through the following courses: (i) SE430 Software Testing and (ii) SE441 Software Quality Assurance.
- 2- All of the main topics of the SWE-KA#10 (Software Quality) contribute to the various learning outcomes of the SE441 course.

5 Discussion and Recommendations

As indicated earlier, this paper checks the compliance of the SWE-Curriculum at JUST with the second five of the fifteen SWE Knowledge Areas (SWE-KAs#6–10) that are indicated in the Software Engineering Body of Knowledge (SWEBOK-V3.0) of the IEEE Computer Society. Table 8 provides an overall view of the

coverage of the SWE-KAs#1–10 (*SWE-KAs#1–5 comes from part 1 [1]*) in the said SWE-Curriculum.

As indicated in Table 8, the compliance of the said SWE-Curriculum with the first ten SWE-KAs can be either Fully Compliant, Highly Compliant, Partially Compliant, or Poorly Compliant.

The previous paper (*P#1*) [1] recommended the introduction of the following two new courses on:

- (a) *Software Construction* (based on Chap. 3 of the SWEBOK-V3.0)
- (b) *Software Maintenance* (based on Chap. 5 of the SWEBOK-V3.0)

Consequently, the Author would recommend the following in this paper:

1. The settlement of the *Software Quality Assurance* course as a core course rather than an elective one in the said SWE-Curriculum.
2. The introduction of the following three new courses on:
 - (a) *Software Configuration Management*: This course is strongly recommended to be based on Chap. 6 of the SWEBOK-V3.0 such that all required Software Construction related topics are covered in it.
 - (b) *Software Engineering Process*: This course is strongly recommended to be based on Chap. 8 of the SWEBOK-V3.0 such that all required Software Construction related topics are covered in it.

Table 6 SWE-KA#9 (SWE Models and Methods) and its coverage in JUST SWE-Curriculum

SWE Models and Methods KA (SWEBOK-V3.0)	Covered in	CLOs Relevance
1. Modelling	SE220	CLOs of the SE220
1.1 Modelling principles	SE220	
1.2 Properties and expression of models	SE220	
1.3 Syntax, semantics, and pragmatics		
1.4 Pre-conditions, post-conditions, and invariants		
2. Types of models		
2.1 Information modelling	SE220	
2.2 Behavioral modelling	SE220	
2.3 Structure modelling	SE220	
3. Analysis of models	SE220	
3.1 Analyzing for completeness	SE441	
3.2 Analyzing for consistency	SE324	
3.3 Analyzing for correctness	SE441	
3.4 Traceability	SE321	
3.5 Interaction analysis	SE440	
4. Software engineering methods		CLO2 of the SE440
4.1 Heuristic methods		
4.2 Formal methods		
4.3 Prototyping methods	SE321/SE440	
4.4 Agile methods	SE440	

Table 7 SWE-KA#10 (SW Quality) and its coverage in JUST SWE-Curriculum

Software Quality KA (SWEBOK-V3.0)	Covered in	Declared in the following CLOs
1. Software quality fundamentals	SE441	CLO1-to-CLO10 of the SE441
1.1 Software engineering culture and ethics	SE441	
1.2 Value and costs of quality	SE441	
1.3 Models and quality characteristics	SE441	
1.4 Software quality improvement	SE441	
1.5 Software safety	SE441	
2. Software quality management processes	SE441	
2.1 Software quality assurance	SE441	
2.2 Verification & Validation	SE441	
2.3 Reviews and audits	SE441	
3. Practical considerations	SE441/SE430	
3.1 Software quality requirements	SE441	
3.2 Defect characterization	SE441	
3.3 Software quality management techniques	SE441	
3.4 Software quality measurement	SE441	
4. Software quality tools	SE441	

- (c) *Software Models and Methods*: This course is strongly recommended to be based on Chap. 9 of the SWEBOK-V3.0 such that all required Software Construction related topics are covered in it.

6 Conclusions

This paper reflects the compliance of the SWE-Curriculum at JUST with the SWE-KAs#6–10 of the Software Engineering Body of Knowledge of the IEEE Computer Society.

The SWE-KA#7 Software Engineering Management and the SWE-KA#10 Software Quality are fully covered (*Fully Compliant*) in the said SWE-Curriculum. While the remaining three SWE Knowledge Areas (SWE-KA#6 Software Configuration Management, SWE-KA#8 Software Engineering Process, and SWE-KA#9 Software Engineering Models and Methods) are partially covered (*Partially Compliant*) in the said SWE-Curriculum. Therefore, the author recommended the introduction of three new courses on (i) *Software Configuration Management*, (ii) *Software Engineering Process*, and (iii) *Software Engineering Models and Methods* into the said SWE-Curriculum. These three new courses come in addition to the two courses that were recommended in the previous part of this research (*P#1*) [1] (e.g., *Software Construction* and *Software Maintenance* courses). Also, in this paper, the author recommended that the existing *Software Quality Assurance* course becomes a core course rather than an elective one in the said SWE-Curriculum.

Acknowledgments The author would like to thank Dr. Ahmed Shatnawi for his valuable input and careful proofreading of the final version of this paper.

References

1. M. Radaideh, Benchmarking the Software Engineering Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE Software Engineering Body of Knowledge (Software Engineering Knowledge Areas #1–5); Accepted in the “*The 18th International Conference on Software Engineering Research and Practice (SERP’20)*”, and in the Research Book Series “*Transactions on Computational Science & Computational Intelligence*” <https://www.springer.com/series/11769>
2. The curriculum of the software engineering undergraduate program at Jordan University of Science and Technology; http://www.just.edu.jo/FacultiesandDepartments/it/Departments/SE/SiteAssets/Pages/Programs/IET-SE_English_StudyPlan2016.pdf
3. P. Bourque, R. Dupuis, *Guide to the Software Engineering Body of Knowledge* (IEEE Computer Society, Los Alamitos, 2004) SWEBOK-V3.0; <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
4. IEEE Computer Society; <https://www.computer.org/>
5. ABET Accreditation; <https://www.abet.org/accreditation/>
6. Institute of Engineering and Technology Site – IET Accreditation; <https://www.theiet.org/career/accreditation/academic-accreditation/>
7. M. Bernhart, T. Grechenig, J. Hetzl, & W. Zuser, Dimensions of software engineering course design, ICSE 2006, Shanghai, China, May 20–28, pp. 667–672 (2006)
8. M. Shaw, Software engineering education: A roadmap. ICSE – Future SE Track, 371–380 (2000)
9. F.P. Brooks, *The Mythical Man-Month, Anniversary Edition* (Addison-Wesley, Boston, 1975)
10. IEEE Std. 828-2012, *Standard for Configuration Management in Systems and Software Engineering* (IEEE, 2012)
11. A.M.J. Hass, *Configuration Management Principles and Practices*, 1st edn. (Addison-Wesley, 2003)
12. CMMI Product Team, *CMMI for Development, Version 1.3* (Software Engineering Institute, 2010).; <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=9661>
13. Project Management Institute, *A Guide to the Project Management Body of Knowledge (PMBOK(R) Guide)*, 5th edn. (Project Management Institute, 2013)
14. Project Management Institute and IEEE Computer Society, *Software Extension to the PMBOK® Guide*, 5th edn. (Project Management Institute, 2013)
15. R.E. Fairley, *Managing and Leading Software Projects* (Wiley-IEEE Computer Society Press, 2009)
16. B. Boehm, R. Turner, *Balancing Agility and Discipline: A Guide for the Perplexed* (Addison-Wesley, 2003)
17. IEEE Std. 15939-2008, *Standard Adoption of ISO/IEC 15939:2007 Systems and Software Engineering—Measurement Process* (IEEE, 2008)
18. J. McGarry et al., *Practical Software Measurement: Objective Information for Decision Makers* (Addison-Wesley Professional, 2001)
19. J. McDonald, *Managing the Development of Software Intensive Systems* (John Wiley and Sons, Inc., 2010)
20. R.E. Fairley, *Managing and Leading Software Projects* (Wiley-IEEE Computer Society Press, 2009)
21. J.W. Moore, *The Road Map to Software Engineering: A Standards-Based Guide* (Wiley-IEEE Computer Society Press, 2006)

22. Project Management Institute and IEEE Computer Society, *Software Extension to the PMBOK® Guide*, 5th edn. (Project Management Institute, 2013)
23. D. Gibson, D. Goldenson, K. Kost, *Performance Results of CMMI-Based Process Improvement* (Software Engineering Institute, 2006) <http://resources.sei.cmu.edu/library/asset-view.cfm?assetID=8065>
24. ISO/IEC 15504-1:2004, *Information Technology—Process Assessment—Part 1: Concepts and Vocabulary* (ISO/IEC, 2004)
25. J.M. Wing, A specifier's introduction to formal methods. *Computer* **23**(9), 8 (1990), 10–23
26. P.B. Crosby, *Quality Is Free* (McGraw-Hill, 1979)
27. W. Humphrey, *Managing the Software Process* (Addison-Wesley, 1989)
28. S.H. Kan, *Metrics and Models in Software Quality Engineering*, 2nd edn. (Addison-Wesley, 2002)
29. ISO/IEC 25010:2011, *Systems and Software Engineering—Systems and Software Quality Requirements and Evaluation (SQuaRE)—Systems and Software Quality Models* (ISO/IEC, 2011)
30. IEEE, *P730™/D8 Draft Standard for Software Quality Assurance Processes* (IEEE, 2012)
31. F. Bott et al., *Professional Issues in Software Engineering*, 3rd edn. (Taylor & Francis, 2000)
32. D. Galin, *Software Quality Assurance: From Theory to Implementation* (Pearson Education Limited, 2004)
33. ISO 9000:2005, *Quality Management Systems—Fundamentals and Vocabulary* (ISO, 2005)
34. IEEE Std. 1012-2012, *Standard for System and Software Verification and Validation* (IEEE, 2012)
35. IEEE Std. 1028-2008, *Software Reviews and Audits* (IEEE, 2008)
36. K. Wiegers, *Peer Reviews in Software: A Practical Guide* (Addison-Wesley Professional, 2001)
37. M. Qiu, L. Chen, A problem-based learning approach to teaching an advanced software engineering course. In 2010 Second International Workshop on Education Technology and Computer Science, 2010
38. V. Garousi, G. Giray, E. Tuzun, C. Catal, M. Felderer, Closing the gap between software engineering education and industrial needs. *IEEE Softw.* (2019)
39. M.C. Bastarrica, D. Perovich, M.M. Samary, What can students get from a software engineering capstone course? in *2017 IEEE/ACM 39th International Conference on Software Engineering: Software Engineering Education and Training Track (ICSE-SEET)*, (IEEE, 2017), pp. 137–145
40. O. Barzilay, O. Hazzan, A. Yehudai, A multidimensional software engineering course. *IEEE Trans. Educ.* **52**(3), 413–424 (2009)
41. S. Dekhane, M.Y. Tsoi. Work in progress—Inter-disciplinary collaboration for a meaningful experience in a software development course. In *2010 IEEE Frontiers in Education Conference (FIE)* (IEEE, 2010), pp. S1D–1
42. K. Daimi, Strengthening elements of teamwork, innovation, and creativity in a software engineering program. *J. Eng. Entrep.* **3**(1), 35–50 (2012)
43. SE230 Fundamentals of Software Engineering; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE230.pdf>
44. SE310 Visual Programming; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE310.pdf>
45. SE320 System Analysis and Design; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE320.pdf>
46. SE321 Software Requirements Engineering; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE321.pdf>
47. SE323 Software Documentation; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE323.pdf>
48. SE324 Software Architecture & Design; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE324.pdf>
49. SE430 Software Testing; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE430.pdf>
50. SE431 Software Security; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE431.pdf>

51. SE432 Software Engineering for Web Applications; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE432.pdf>
52. SE440 Project Management; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE440.pdf>
53. CS318 Human-Computer Interaction; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/CS318.pdf>
54. SE471: Client Server Programming; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE471.pdf>
55. SE326: Software Engineering Lab; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE326.pdf>
56. SE441: Software Quality Assurance; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE441.pdf>

Moh'd A. Radaideh is currently an *Associate Professor* with the Department of Software Engineering, Jordan University of Science and Technology. He is a Senior Member of the IEEE, the IEEE Computer Society, and the IEEE Education Society. He received his BENG & MENG Degrees in Electrical and Computer Engineering from Yarmouk University and Jordan University of Science and Technology, consequently in 1987 and 1989. He obtained his Ph.D. degree in Electrical and Computer Engineering (Software Engineering) from McMaster University (Canada) in 2000.

Profile: http://www.just.edu.jo/admissionuploads/staff_cv/maradaideh.pdf.

Part II
Educational Tools, Novel Teaching
Methods and Learning Strategies

Design for Empathy and Accessibility: A Technology Solution for Deaf Curling Athletes



Marcia R. Friesen, Ryan Dion, and Robert D. McLeod

1 Introduction

This work presents a redeveloped introductory course in Digital Systems Design, often considered a course on embedded systems and IoT, in an undergraduate computer engineering program at a Canadian university. The course was moved from a theory-driven course to a team project-based course that expanded its technical focus to encompass a new design paradigm, an intentional focus on design for accessibility—in this case, for deaf curling athletes—and an augmentation with knowledge transfer topics around the engineering design tasks. This chapter describes the course and the motivations for its redesign.

2 Background

The course, ECE 3760 Digital Systems Design 1, is a required course in the undergraduate Computer Engineering program. It typically enrolls 20–30 students, and it has historically focused on embedded systems design and an increasing emphasis on IoT. As a student asked, though, “what is *not* considered an embedded system these days?”

The course was redeveloped in 2019–2020 in light of program accreditation requirements that continually emphasize design alongside theory in the curriculum, for opportunities for students to develop teamwork, communication, and other

M. R. Friesen · R. Dion · R. D. McLeod (✉)

Electrical and Computer Engineering, University of Manitoba, Winnipeg, Canada

e-mail: marcia.friesen@umanitoba.ca; umdion@myumanitoba.ca; robert.mcleod@umanitoba.ca

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_8

103

professional skills, and with the support of the NSERC Chair in Design Engineering in Sustainable Design.

In its redevelopment, the course was purposely built around a central design project with both group and individual elements for students, and the design project was chosen as an assistive technology product.

2.1 Design Project: Technology Solutions for Deaf Curling Athletes

In the sport of curling, a team of four throws a total of eight rocks down the curling sheet playing against another team of four (and their eight rocks), with the objective to land as close to the center (“button”) of the target (“house”) at the other end of the sheet of ice. The trajectory of the curling rock is determined both by the release of the player throwing the rock and by the sweeping of one, two, or all three other team members of the ice in front of the rock as it travels down the 156 ft ice sheet. Sweeping in front of the rock can extend its range and change its trajectory, or both, and it is a critical skill in the game of curling. Who sweeps, when to sweep, when to stop sweeping, and how hard to sweep are real-time decisions called by the skip (team leader) as the rock is travelling down the ice.

In most games, the sweepers can listen for and hear the skip’s instructions as they walk or slide down the ice alongside the rock in play. Once sweeping, the player will often be exerting considerable force through their shoulder and arm to sweep, requiring their full attention.

Deaf curlers are at a considerable disadvantage when it comes to sweeping, because they have to constantly look back and forth from the rock/ice to the skip at the end of the sheet to receive visual signals to sweep or not sweep. In relation to hearing curlers, deaf curlers receive instructions on a delay, they lose power in their sweeping position as they are rotating their torso and head back and forth from the ice to the skip, and they are also at higher risk of “burning” a rock (taking it out of play by touching it with their broom).

The class was challenged to design a technology solution for deaf curling athletes that would allow them to receive sweeping signals from the skip and that would allow them to keep their eyes on the rock as they travelled down the ice sheet.

2.2 Conceptual Frameworks

Several conceptual frameworks were considered together to guide the course development, including sustainable design, a modified design cycle that explicitly incorporates empathy, and design for accessibility.

Sustainable design may be considered under the broader umbrella of sustainable development. Sustainable development may be defined as economic development, social inclusion, and environmental sustainability, all enacted under good governance. It has also been defined as development that meets the needs of the present without compromising the ability of future generations to meet their own needs. Sustainable design is design that is aligned with sustainable development principles, and it includes attention to life-cycle low-impact materials, life-cycle energy efficiency, reuse and renewability, biomimicry, new relationships between people and products, and social equity outcomes that are inherent in the UN Sustainable Development goals [1–4].

The second conceptual framework for the course is the engineering design cycle [5]. There are many visual representations of the well-known design cycle (Fig. 1), which is typically introduced in the first year of an engineering program and practiced in courses throughout the curriculum, often culminating in a senior-year capstone design project. Although the choice of words may differ, the familiar engineering design cycle typically includes:

- Defining or identifying the problem
- Requirements gathering
 - Functions, objectives, constraints
 - Stakeholders: Users, operators, clients, others
 - Info gathering: codes, standards, etc.
 - Background research
- Generating ideas, brainstorming, sketching, discussing
- Decision-making: selecting an idea, building, prototyping
- Testing: evaluating, analyzing, checking
- Iterating, re-designing
- Communicating, sharing

While this design process has indeed led to world-changing and life-changing technology solutions, it has not always hit the mark. The literature is replete with examples where design has not considered its users fully, or has discounted the needs of some users.

The examples range from the original automobile crash test dummy being modelled on the 90th percentile Caucasian male body height and weight, as male muscle mass distribution, bone density, vertebrae spacing, and body sway. It was only in 2011 when a female crash test dummy began to be used in testing in the United States. As a result of these design choices, when women are involved in a car crash, they are 47% more likely than men to be seriously injured, 71% more likely to be moderately injured, and 17% more likely to die even when researchers control for factors such as height, weight, seatbelt usage, and crash intensity [6].

Other examples of incomplete design are as diverse as they are hidden in plain sight. The formula to determine standard office temperature was developed in the 1960s around the metabolic resting rate of the average man. A Dutch study found that the metabolic rate of young adult females performing light office work is

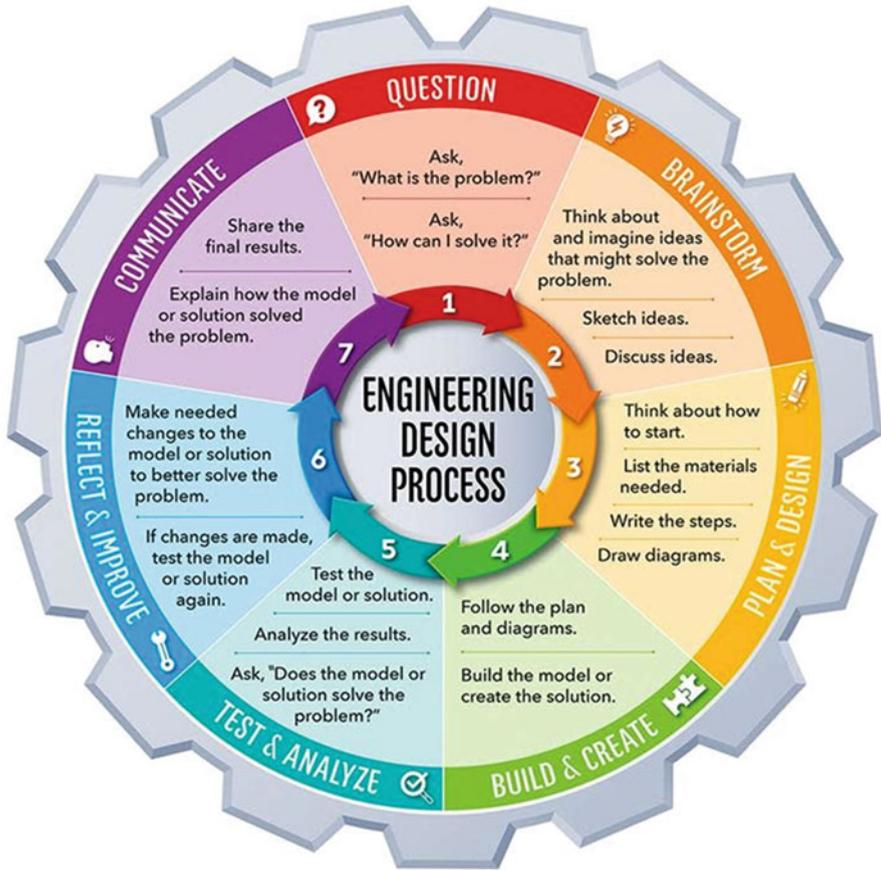


Fig. 1 Typical engineering design process. (Source: <https://www.learningtreecanada.com>)

significantly lower than the standard values for men doing the same activity. In fact, the formula may overestimate the female metabolic rate by as much as 35%. Thus, current offices are on average five degrees too cold for women, leading to lowered productivity due to discomfort. In many jurisdiction, employers are legally required to provide personal protective equipment (PPE) on the job. Most PPE is designed for the sizes and characteristics of male populations from Europe and the United States. Women are often “accommodated” by giving them the smaller sizes of suits, goggles, harnesses, vests, etc. However, the “standard” US male face shape for dust, hazard, and eye masks means they don’t fit most women, as well as lots of black and minority ethnic men. In protective services, women often report that PPE like body armor, stab vests, and jackets did not protect them adequately and/or allow them to do their jobs effectively due to differences in body proportions of chest, hips, and thighs between women and men. Users reported injuries from the PPE

Stanford d.school Design Thinking Process

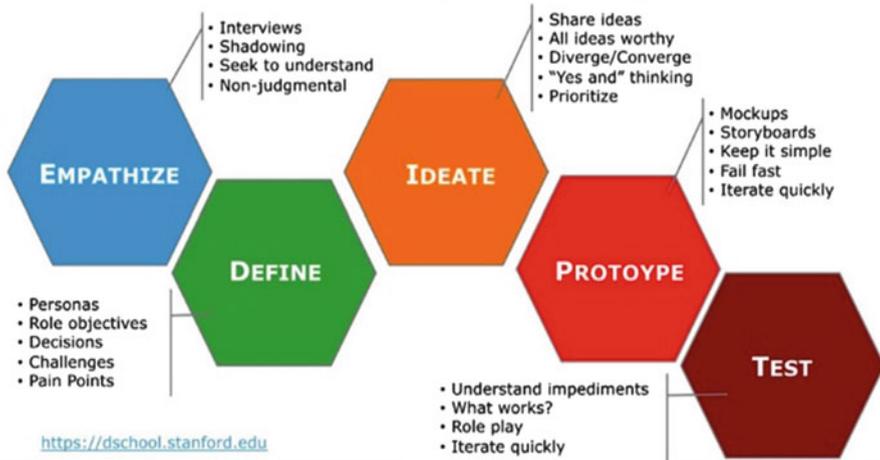


Fig. 2 Stanford d.school design thinking. (Source: <http://www.theagileelephant.com>)

itself and from the PPE not serving its purpose properly on the bodies of women and “nonstandard” men [6].

Originating with Hasso Plattner, Christoph Meinel, Larry Leifer, and colleagues and carried forward by the d.school at Stanford University [7], a modified design cycle is proposed (Fig. 2) which explicitly introduces *Empathize* as the first step, that is, learning about the audience for whom you are designing. This involves nonjudgmental processes of listening to users with a focus on understanding stories and the emotions behind them. In the course, Stanford’s model of design thinking and social in the definition of Sustainable Design come together in the project to design a technology solution for deaf curling athletes.

Equity is understood as one strategy to achieve fairness. Another strategy would be equality, in which everyone is given the same thing (sameness). However, equality can only produce fairness if everyone starts from the same place and needs the same help. Equity is giving everyone what they need to be successful or giving everyone access to the same opportunity.

The third conceptual framework brought to the course is the HAAT model of assistive technology design [8]. The HAAT model considers the human, the activity, and the assistive technology. These considerations include the human’s abilities and skills across cognitive, physical, and affective domains. It also considers the physical, cultural, social, and institutional contexts within which the assistive technology is used, including such qualitative features as stigma associated with some forms of assistive technology.

3 Course Design

Built around the project of designing a technology solution for deaf curling athletes, the course was designed to flow through a range of topics. These included:

- Conceptual stances: Engineering design processes and stages; ethics and equity in design; and design of assistive technology
- Technical considerations in their projects: Wireless communication options; networking; security; power options; user interface options and ergonomics; and component integration
- Functional and physical prototyping
- Allied topics for knowledge transfer: Business development, technology marketing, intellectual property protection, and moving toward commercialization to build awareness of allied professionals with whom engineers will work or areas into which their own careers may advance

The first and fourth topics above represented the major departure from the historical delivery of this course in the curriculum. These topics were advanced over a traditional 13-week term (3 hours/week plus laboratories) by two co-instructors and augmented by content area experts as guest speakers. The associated assignments were as follows:

- Specifications document for the design (one per student and then a revised document as one per team).
- Choice of a marketing plan or a business plan for the design (one per team).
- Provisional patent application for the design (one per student).
- Two-minute elevator pitch or design review (depending on design stage).
- The final project components consisting of a functional prototype of the design and physical 3D model of the sweeper and skip interface (one per group), a conference-style poster of the design (one per group), a group presentation of the design (one per group), and a final reflection piece on the design and the course.
- These formal assignments were augmented by regular (1–2 times/week) “short snappers,” that is, assignments that would take 10–30 minutes to complete and reinforced the day’s or week’s materials. These collectively offered 5% extra credit to students who completed them. Topics included curling rock physics, flowcharts, state charts, message sequence charts, activity diagrams, bit addressing, brain teasers, and examples of poor designs students have encountered.

Several elements intentionally highlighted the *Empathize* stage of the modified design process. In the specifications assignment, the first version was written individually by each student. Traditional design process criteria for specifications were required, including functions (verbs), objectives (adjectives), constraints, and documenting the context, users, and stakeholders. This was done before the first lab session (described below). After the first lab and now as a design group (4–5 members), a revised specification document was written as a group document. In

the second version of the specifications document, students were guided to write in an appreciative framework, rewriting negative statements like “it must not . . .” or “it cannot be . . .” as “it will . . .” Further, top-level functions were re-framed into human-centric problem statements, that is, to write from the curler’s point of view, not the device’s point of view. Statements like “the device should . . .” were re-written to “the curler should be able to . . .” or “the curler will . . .” The revised specifications were then provided to a deaf curling team for their feedback.

Further, the first lab in the course (of five labs overall) consisted of the class going to a local curling club to observe the physical environment (size, temperature, lighting, social interactions, etc.) as well as learning and playing the game. For all but two students, it was the first time playing the game. In the laboratory report for the first lab, students reflected on their observations of the environment, the game, and the various stimuli and demands on the players. These observations ranged from the size of the rink and the length of the ice sheet, the low temperature in the building, the noise levels, the lighting, social interactions on neighboring ice sheets, interactions between team members in a given game, the duration of a single play, the clothing worn during play, the hardness of surfaces, the networking options within an arena, and more. Students were also asked to explicitly consider how a deaf curler would experience the game, and some students brought earplugs to assist in that exercise.

The other four labs introduced students to the TM4C123GXL LaunchPad Development Kit by Texas Instruments and the CC3100 WiFi BoosterPack. These were selected as they are fairly well entrenched in the embedded systems community with considerable online support, particularly from J. Valvano at the University of Texas at Austin. Some of the groups suggested the PIC32 or Arduino as alternatives. Although these are reasonable suggestions, we also wanted the groups to use hardware and IDEs that they had not yet encountered. The first hardware lab was an introduction into the Keil IDE and simple experimenting with LEDs and establishing a WiFi connection. The second lab introduced switches and timers. The third and fourth labs introduced a more integrated design that would functionally emulate the design the groups had come up with. In addition to the LaunchPad and BoosterPack, the lab kit included rudimentary switches and LEDs.

Several of the groups had also considered incorporating patch vibrators into their design, for sweeper notification and accelerometers as part of the skip’s controller. As this was the inaugural year of the design project, we had not anticipated inventiveness and novelty of the student designs. Thus, if employed in functional prototypes, their function had to be emulated. In the coming year, we will include considerably more options within the lab kit in an attempt to meet the inventiveness of the student designs.

4 Students' Designs

The students, working in groups of four to five members, created five designs that all consisted of a controller for the skip to communicate sweeping instructions to the sweepers and a device by which the sweepers will receive and perceive the instructions. The essential instructions are to either sweep or not sweep, and the next level of complexity is that if sweeping, whether to sweep “normally” or whether to sweep hard. However, in even recreational league play, the skip will also want to communicate whether they want one or two sweepers to sweep, and, if only one, which one (usually identified by being to the right or left of the rock as it travels down the ice).

Some of the combinations devised by students consisted of:

- The skip using a handheld remote with buttons to communicate sweeping instructions to a LED bank attached to either the face or handle of the sweepers' brooms
- The skip using an app on their phone to communicate sweeping instructions to a LED bank attached to either the face or handle of the sweepers' brooms
- The skip using a glove-like or handheld controller with gesture, inferencing from an accelerometer device to use hand and arm movements to communicate sweeping instructions to the sweepers
- A button-based skip controller to a vibrating motor(s) embedded in an undershirt worn by the sweepers

In each case, the communication between skip controller was over WiFi as that was what the BoosterPack provided and would meet any range requirements. Most designs used a phone as a WiFi hotspot effectively serving as an access point.

For students using an LED bank to represent sweeping instructions, their combinations included solid and flashing lights to signify sweeping (e.g., solid green to sweep, flashing green to sweep hard, and solid red to stop sweeping) and various configurations for signaling to only one or the other sweeper, or both. Two student teams also considered red-green color-blindness for the sweepers and designing the LED device to use physical positioning of the LEDs to create separation between red and green LEDs.

As they refined their prototypes, students were confronted with a number of physical and ergonomic issues, including how a remote held by the skip needs to be quickly accessible after the skip throws their own rock (and is holding their own broom in their other hand), intuitive button placement on a remote that follows natural finger and thumb movements, whether flashing LEDs are perceptible when sweeping fast, and whether vibrations on the body are perceptible when moving and wearing other heavy clothing.

Technical considerations included the wireless communication options (or limitations) in an arena, durability of devices used on an ice surface, and battery life in an ambient temperature of -3°C (27°F) in a curling arena. In addition, most students recognized a variety of alternatives for the wireless connectivity which may

Fig. 3 Rendering of the phone sleeve physical model for the skip



be preferable, for example, BLE between the skip's controller and their smartphone as being preferable to WiFi. If a smartphone were to serve as the skip's controller, consideration had to be given to ergonomics: how could one ensure that it were being used correctly while the skip is preoccupied with watching the rock and not able to look at the phone while providing their sweeper directions?

It was also a very valuable exercise to have the two-minute design review or elevator pitch (depending on the teams' design stage) as students were able to discuss design choices made by others and incorporate some of the ideas into their own design. The most significant of these were resolving how the skip would communicate to two or three sweepers who could arbitrarily be on either the left or right side of the rock.

The following are some of the physical design models suitable for 3D additive manufacturing. The first (Fig. 3) is a skin that would fit over a phone and facilitate tactile control. In this particular design, the user can differentiate between the left and right sliders via a tactile guide across the middle of their screen, without having to look at the screen.

A second aspect of the design is the broom attachment. Most of the groups settled on something similar to that of Fig. 4. As in the example in Fig. 4, the designs had holes for red and green LEDs in some configuration as well as means or affixing the device to the broom. Only one of the groups had a provision on the broom attachment for selecting the left or right side of the rock for sweeping.

We asked the students to consider a physical prototype for several reasons, including evaluating the functionality and ergonomics of their design in real use. Figure 5 illustrates a design that would be attached to the broom that the skip would hold. In Fig. 5, the controller is attached to the broom (potentially with a Velcro strap and rubber pad to hold in place, attachment method pending). It is meant to ergonomically fit the hand. Two two-step push buttons actuated with the index finger are used to control the two sweeper channels.

Figure 6 was one of the more comprehensively thought-out designs that captured the initial statechart (Fig. 7) of the design for that group. The statechart (Fig. 7) was the initial design entry point for refining functionality of both the skip controller and the broom devices. These were student-driven with little or no feedback from the instructors as we did not want to unduly influence the design process through group-

Fig. 4 3D model of the broom attachment

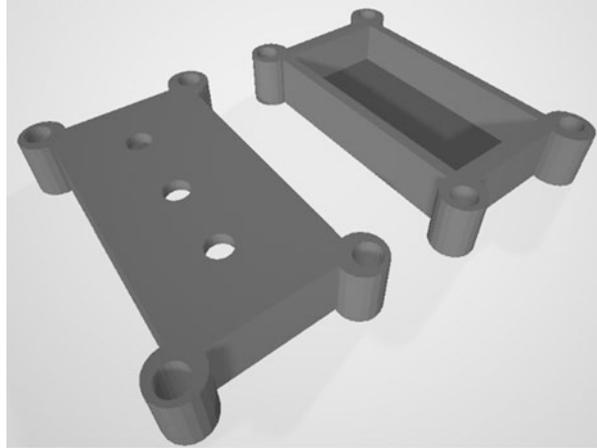


Fig. 5 3D model of the skip controller



think. General feedback was provided, such as “ensure your design can account for two out of three sweepers being in either the left or right side of the rock.”

As instructors, we provided feedback to each group from the perspective of being curlers, although we not deaf. While our clients, the deaf curling team, were very interested to remain involved in email communication with the instructors, they were not comfortable communicating directly with students via email nor attending in class with ASL interpreters. We also tried to serve as proxies for deaf curlers in trying to keep the designs user-centric. For example, having a design like Fig. 5, while likely ergonomic, is attached to the skip’s broom. However, while the skip is throwing their own rocks, the control or line calling defaults to the vice-skip or third.

Fig. 6 3D model of the skip controller with “left,” “right,” and “both” switch as well as sweeping vigor control

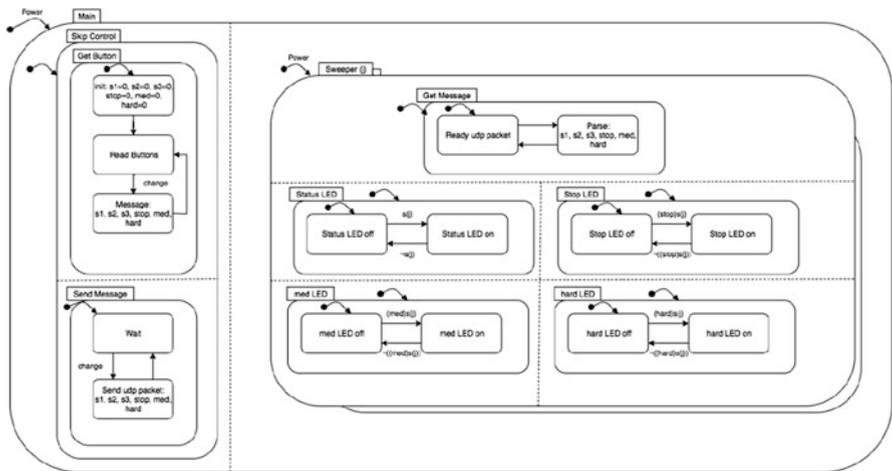


Fig. 7 The initial statechart from which the skip controller physical prototype of Fig. 6 was based

5 Discussion and Conclusion

Overall, as instructors, we were pleased with the first iteration of the course and with students’ receptiveness to the course. It is well-known that students build perceptions of courses by asking friends who have previously taken the course, and it was apparent that students perceived this to be a significant departure from the course in previous years. It is also well-known that attendance can be one indication of engagement, and we were pleased to see very strong attendance throughout the term. Because of the degree of new material presented as well as the degree of nontechnical material presented, it is incumbent upon us as instructors to be especially mindful to provide students with ongoing guidance of how each piece fits

into the whole and to recognize the level of intellectual work it takes to integrate a varied range of information for the first time. In line with good teaching and learning practices, we began each class with a degree of “wayfinding” to remind students where we had been, where we were going, and where the present class fit into the overall map of the course. This appeared to be important to ease anxieties about a course and project that stretched students in several areas.

It is also clear that for a course such as this with an actual product design in mind, the hardware materials available to the students have to be flexible, in this case, for example, inclusion of accelerometers and vibrators. In addition, although we were focusing on functional and physical prototypes, in the future, additional emphasis will be on low-power and power-saving modes. As we also want to continue with the same design for 2–3 years, we will shift toward incremental design improvements as opposed to the *tabula rasa* of this year. Future designs should also place greater emphasis on security and the use of IoT frameworks such as MQTT.

We were pleased to have the opportunity to integrate design for accessibility and equity into the course in a meaningful way. Both instructors are highly interested in curling, and although none of the students were curling athletes, they built an appreciation for the sport and in particular for the challenges facing deaf curlers over the duration of the course. Because the term coincided with major national and international curling competitions, there were a myriad of ways to highlight how the design was not in any way a “toy problem” but rather a niche awaiting entrepreneurial ideas. That being said, it is also the case that any modifications to athletic equipment used in competitive play would require vetting and approval by the sport bodies, such as Curling Canada.

In future iterations of the course, we will explore ways to engage the clients with the students directly while respecting the clients’ abilities and desires. In this case, the deaf curlers were pleased to receive email information from instructors but were very reticent to engage directly with students.

One of the final assignments in the course (ending April 7, 2020) is for students to write a reflection piece. They are given a wide berth in this reflection, encouraged to comment on what went well in the project, what did not go well, and what they would do differently if they had to do it again. Students were encouraged to think of these first three questions from a technical point of view, design and team processes, social dynamics, integration of multiple objectives of equity and accessibility, client perspectives (in this case, deaf curlers), etc. Students were also encouraged to reflect on their key takeaways or learnings from this process (positive insights and difficult realizations alike), what they are proud of in the process, and what else sticks with them about the project and the process.

In their reflection pieces, students indicated that the opportunity to design something from conceptual design to completed prototype was a welcome task, as many felt it eased their nervousness for the senior capstone design course in the coming year by giving them a “practice run.” Students also found motivation in working on a design problem with real-world impacts for people with disabilities. Students also indicated that including allied elements of designs was eye-opening

to them, for example, a provisional patent application and the choice of a business plan or marketing plan.

Using the first lab period to go curling was well received, and most said it was extremely helpful to understand the problem better. Many commented on the differences in watching curling videos as opposed to being present in the curling club and noting the physical environment. Because of most students' lack of familiarity with curling, spending time generating ideas and vetting ideas as a class was very useful.

Like most North American colleges and universities, the academic term was cut short due to the COVID-19 pandemic. Very abruptly on March 12, 2020, our institution ceased in-person classes, and the final 3 weeks of the 13-week course had to take place remotely and online. While this hampered some teams' abilities to take their design to a final physical and functional prototype, all teams were able to demonstrate their design as proof-of-concept, and all submitted their final presentations by video and e-poster. The students' creativity, resilience, and ongoing enthusiasm are to be highly commended.

Acknowledgments Many thanks to the 22 students in ECE 3760 Digital Systems Design 1 at the University of Manitoba in Winter 2020 for their enthusiasm and dedication to a new course format and focus. Special thanks to the invited guest lecturers including Dr. Danny Mann (design for accessibility), Mr. Jem Burkes (security and low-power design consideration), Mr. James Dietrich (wireless communication options), Dr. Ken Ferens (ARM bit specific addressing and interrupts), Mr. Jay Diamond (business development and technology marketing), Mr. Blake Podaima (provisional patents for entrepreneurs and technopreneurs), and Dr. Douglas Buchanan (intellectual property protection within a multinational corporation). Finally, many thanks to the Lavallee curling team at the Heather Curling Club for inspiring this project and for working with us through the duration.

References

1. J.D. Sachs, *The Age of Sustainable Development* (Columbia University Press, New York, 2015)
2. Report of the World Commission on Environment and Development: Our common future. Available: <http://www.un-documents.net/wced-ocf.htm>
3. B.A. Striebig, A.A. Ogundipe, M. Papadakis, *Engineering Applications in Sustainable Design and Development* (Cengage Learning, Boston, 2016)
4. UN sustainable development goals, <https://sustainabledevelopment.un.org/?menu=1300>
5. S. McCahan, P. Anderson, M. Kortschot, P.E. Weiss, K.A. Woodhouse, *Designing Engineers: An Introductory Text* (Wiley, 2015)
6. C.C. Perez, *Invisible Women: Exposing Data Bias in a World Designed for Men* (Random House, 2019)
7. Stanford d.school, <https://dschool.stanford.edu/>
8. A.M. Cook, S.M. Hussey, Assistive technologies, in *Principles and Practice* (2002)

An Investigation on the Use of WhatsApp Groups as a Mobile Learning System to Improve Undergraduate Performance



A. Rushane Jones and B. Sherrene Bogle

1 Introduction

In this technology-driven world, there is no doubt that the Internet plays a vital role in information exchange between individuals and businesses [1]. The widespread use of WhatsApp among university students and the frequency of its use suggested that it could be used as a nonintrusive learning platform and would bolster the educational potential of this predominantly social platform [2–4].

Some of the major benefits of social network sites are that they are inexpensive, easy to use, convenient, and entertaining, as long as there is reliable Internet and electricity. Their use doesn't require obtaining costly hardware tools such as laptops and PCs and can be used to motivate students, replace learning management systems, and assist with communication between instructor and student [2, 3, 5].

Research has proven that the use of social media can improve the academic performance of students in higher learning by providing a familiar online environment that is directed at tertiary education [2, 4, 5]. The goal of the work within this study is to assess the creation and use of a WhatsApp group that allows students to engage in a more sociable class setting. The aim is to promote growth and discussions beyond the confines of the typical classroom, in order to boost academic performance and interest.

A. R. Jones (✉)

School of Computing, University of Technology, Kingston, Jamaica
e-mail: rushane.jones@utech.edu

B. S. Bogle

Humboldt State University, Arcata, CA, USA

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_9

117

1.1 Background

The island of Jamaica, a developing country, is home to 2.934 million people and has five recognized universities and several colleges island-wide, each with diverse courses and learning methodologies [5–7].

Social media sites have had exponential growth since their inception. WhatsApp, which is currently the most popular mobile messaging app, boasts 1.5 billion monthly users in 180 countries, with its stablemate, Facebook Messenger, being 0.2 billion users behind which accounts for the predominant use of mobile devices among undergraduate students [8].

The average college student spends approximately 3 hours per day on WhatsApp [9].

The participants of this study were animation students in their freshman and sophomore years. Students participated in a blended mode of delivery where the instructors of the respective courses used WhatsApp groups as a learning management system and reported higher levels of motivation and class engagement. The conceptualized framework made use of the second thesis of Anderson's Interactivity Equation in order to investigate the major influences of motivation toward learning and why this phenomenon occurred with students who were more active in the class while using the created WhatsApp groups [3, 5, 10, 11].

The most common teaching methodology in higher education takes place in a classroom setting with lectures, tutorials, and practical classes creating different settings, such as quizzes and discussions group activities, and in some cases, students are referred to a Learning Management System to interact with [12–14].

1.2 Contribution

Several studies have embarked on the use of WhatsApp by students as a means of an e-learning environment in developing countries. There have been studies on how the use of WhatsApp affects students in higher education, but very few look at the deliberate use of instructor-driven WhatsApp groups in improving academic performance of tertiary students in developing countries.

The students and faculty of the National University will always be introduced to new technology as time progresses, and so it is important to prepare ourselves for it. The stakeholders need change their thinking in how technology made for a specific reason can be used for a beneficial purpose in other areas, as is the case here for education. To address the aforementioned gap in the literature, this paper aims to show how the Interaction Equivalency Theory can be used to explain and predict academic performance in university students who do practical courses. This study utilizes WhatsApp as a nonintrusive learning environment for students and peer collaboration, in order to improve student learning outcomes, and which further

challenges the way curricula are delivered in higher education. This builds on a previous paper done using Facebook as the mobile learning environment [5, 15].

1.3 Research Question

The following question was used to guide this paper:

What factors affect the relationship between social media interaction and academic performance?

1.4 Hypothesis

H0 – Tertiary-level students would not improve in their academic performance through the use of WhatsApp groups as a LMS.

H1 – Tertiary-level students would improve in their academic performance through the use of WhatsApp groups as a LMS.

1.5 Variables

There were one dependent variable and three independent variables. The dependent variable was academic performance, while the independent variables were student-teacher interactions, student-student interaction, and student-content interactions.

1.6 Limitations

Convenience sampling was used in part because of the proximity of the location as well as the availability of the participants.

2 Literature Review

2.1 Social Media in Higher Education

Higher learning has often been an area of frequent research, especially in topics concerning the improvement of academic performance. This section discusses

related studies that point toward the gap as well as theoretical framework used in the study.

A study was conducted by [5] to test whether or not WhatsApp groups could be used as a learning management system to improve academic performance. The Interactivity Equivalency Equation was used to ascertain the impact of WhatsApp groups on student outcomes. Significant factors were the value placed on real-time chatting with peers, practical assessment on overall learning, and the reading of instructor posts. 75% of the control group had GPA less than 2.70, which suggests that the lack of the additional interaction may have been a contributing factor to lower GPAs than the experimental group which had a mean GPA of 3.71.

2.2 Learning Management Systems

The LMS has become an integral part of the educational system in most universities, and interest has continued to grow in combining the traditional teaching methodologies with online activities to create blended modes of teaching. “LMS are not meant to completely replace the traditional classroom setting but is meant to supplement the traditional lecture with course content that can be accessed from campus or the Internet” [13, 16]. Social media sites have become dominant for education and entertainment. As humans, we are akin to forming communication with people of similar interest. In education, there are primarily two trains of thought prevailing when it comes to social media in education. The first outlook is that social media can be beneficial as a tool in supporting activities deemed important by educational institutions, instructors, and students. The second outlook is that social media has a bad influence on student performance and inflicts poor behavior and time management in students [17, 18].

2.3 WhatsApp in Higher Education

A study was conducted to investigate whether students’ perception that WhatsApp was useful shaped the cognitive process needed for teamwork. The study included a sample of 200 university students that were engaged in a role play where the use of WhatsApp had played a major function as the communication tool and students were asked to work in teams to solve decision-making assignments. The results of the study had confirmed that the perceived usefulness of WhatsApp and the team’s efficacy had a correlation and produced better grades. This study also noted that WhatsApp is the most popular among the typical undergraduate age group [19].

A study that was conducted in Pune, India, with 138 respondents, which sought to find the preference of use of WhatsApp in colleges, find awareness and availability of WhatsApp groups, and whether all the ethical rules were being applied to its use, especially in the area of sending fake messages. The author had written that

the use of WhatsApp made it easier to communicate with people and was used to enhance discussions and sharing information among students and lecturers. The study determined that 89.9% of respondents had a preference for WhatsApp groups noting it to be a good time-saver and a good communication medium as it is fast and a lot of information could be shared; the remaining 10.1% had no preference because it could be misused by students, there is a lack of confidentiality, and that it was unethical to be used. The author concluded that from an ethical standpoint, there existed gaps in understanding about not forwarding fake messages [10].

Reference [11] conducted a study using WhatsApp as part of a blended learning model to teach programming, highlighting that the problem was caused by the slow pace of learning, difficulty in grasping syntax, and poor logical skills. The notion behind the study was to take advantage of the popularity of WhatsApp among the students and use it for educational purposes in order to help programming novices. Although the implementation was not complete, the technical guidelines took note of limitations of the app and also laid down rules for proper use of the group and the benefits. The study concluded that the use of social media is on the rise and that higher education should embrace it, not as a means to replace the already existent courseware infrastructure but as a means of supporting, modernizing, and socializing the learning of programming.

In [20], that study explored the use of social media in the context of communication and learning in higher education and the difficulties by using them. Data was retrieved quantitatively and qualitatively through survey distribution to many facilities. The results identified WhatsApp as the most popular social media platform and a priority for communication among students, especially for the development of learning applications. The issues that the students faced by using social media were misunderstanding and a lack of Internet connection. Some of the recommendations arising from the study were the development of future social network features that offer ease of use in general communication as well as links for learning and greater storage limit.

2.4 Theoretical Framework

The Interaction Equivalency Theorem used in [5] proposed by [21] “aims to clarify how interaction works in distance education. It proposes an argument that there is a difference in schemes between the independent versus interactive oriented distance learning activities and these need to be taken into consideration when designing and delivering distance education that meet the needs of its learners.”

Figure 1 shows a visualization of the Interaction Equivalency Theorem. Thesis 1, which is likely to be associated with a closed system, stated that there would be deep and meaningful learning if one of the three forms of interactions were performed at a high level. Thesis 2, which is likely to be associated with an open system, stated that there needed to be a high level of at least one of the three types of interaction, for a more engaging learning experience [5, 21]. A closed system is usually defined

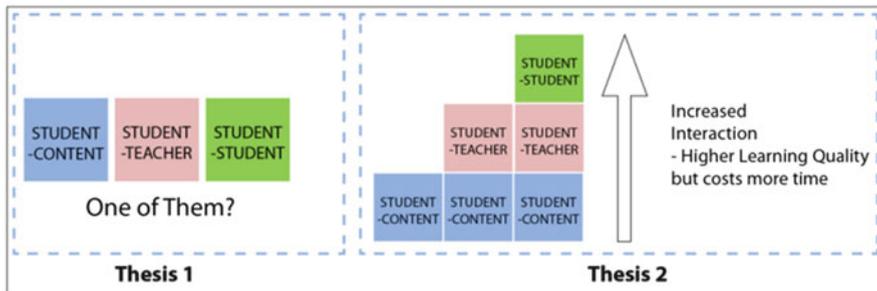
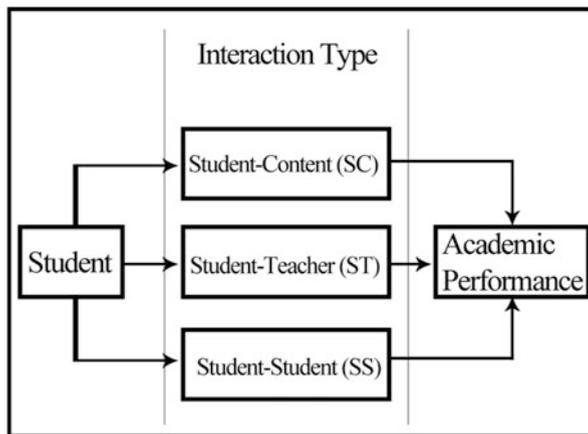


Fig. 1 Interaction equivalency theorem [5, 21]

Fig. 2 Conceptual model based on Thesis 2 of the Interaction Equivalency Theorem [5, 21]



by a system of learning where usually one type of high level of interaction occurs and is stipulated by design, to facilitate efficient and effective learning, at a minimal cost. Thesis 2 however is usually defined by an open system where multiple types and levels of positive surplus interactions occur, such as impromptu site visits or guest speakers. While it is not impossible for a course to be planned in a manner that has three high-level interaction, that would warrant the need of time and cost.

The conceptual framework seen in Fig. 2, based on [21], models the framework for this continued study.

It was used in order to improve the academic performance of the students involved in the study. Based on the literature, three constructs, student-content interaction (SC), student-teacher interaction (ST), and student-student (SS) interaction, exist in the Interaction Equivalency Theorem. The conceptual map shown in the figure above delineates how academic performance was affected in the study.

2.5 Gaps in the Literature

Firstly, very few studies have explored the use of WhatsApp as an alternative means of creating a collaborative learning environment that is managed by the instructor. Secondly, there has been very little formal research done on the use and implementation of social media and higher education in the context for a developing country. Thirdly, there is limited literature indicating whether there is a definite association between social media learning environment and its impact on academic performance, especially in tertiary education. To address the gap, this study demonstrates how the Interaction Equivalency Theory can be used to explain and predict academic performance in university students who do practical courses while using statistical methods to make inferences and identify correlations.

3 Methodology

3.1 Purpose of Research

This research aims to improve on the practice of using social media technology and its features for academic purposes and improving student learning outcomes, including grades. This research hopes to change the culture of faculty and promote the early adoption of new technology. It investigated the correlation of student grade improvement through the use of class-based WhatsApp groups, with the primary focus of sharing resources, class material, as well as doing revisions, and answering queries and periodical reminders of assessments.

3.2 Research Design

This study used a case study approach using quantitative and qualitative data analysis [5, 22]. This was chosen as to explore the causality of the varying levels of interaction in a practical tertiary classroom setting and determine how it could shape the interaction of multimedia-based courses.

3.3 Population and Sample

The sample for the case study was taken from two (2) classes of students within animation majors at the national university. The population of students at the institution is approximately 13,303. Approximately 1220 students attend the School of Computing and Information Technology (SCIT) where the study takes

place. Each occurrence of the modules in the study was capped at a maximum of 25 students per session.

The participants of the study enrolled in the respective modules Critical Structures (Storyboarding Concepts), Dynamic Anatomy 1, and Animation Tools I which were offered in a blended method of teaching. This study was limited in scope to investigating interaction experiences and preferences of these learners in order to provide in-depth data about specific interactions with technology and how it affected their academic performance.

3.4 Sample Course Description

Students were taught in a blended mode of study and met with the instructor for a minimum of 4 hours per week. For each module, WhatsApp groups were created to maintain dialogue between the students outside of class hours. Assignments, objectives, and notes were given to the students via the group. At the end of the semester, the results of the assessments were collated, and the final grade was generated.

3.5 Instrument for Data Collection

The instrument used in the study was an online survey distributed through WhatsApp groups. The questionnaire was created and distributed through Google Forms and consisted of open-ended and closed-ended questions adopted from [5, 23, 24].

Based on Cronbach's reliability analysis, 30 close-ended questions were used in order to meet a good standard of reliability [25]. The questions gathered data on demographics and the SC, SS, and ST constructs. The constructs were tested using a 5-point Likert system. The variables of Cronbach's Alpha analysis for the variables SC, SS, and ST were 0.73, 0.83, and 0.70, respectively. Additionally there were a few open-ended questions to garner feedback on the student's experience. Table 1 summarizes the instrument.

Table 1 Summary of instrument

Section	Items	Relates to	Item labels
1	1–8	Demographic information	
2	9–15	Student-content	SC00, SC01, SC02, SC06, SC07, SC09, FB_SC01, ST04
3	16–23	Student-student	FB_SS01, FB_SS02, SS01, FB_SS03, FB_SS04, FB_SS05, SS04
4	24–30	Student-teacher	FB_ST01, FB_ST02, ST02, ST03, FB_ST03, FB_ST05, ST09

3.6 Data Preparation and Preparation

Much like the study in [5], each item in the questionnaire was given a variable name, and the responses for each question were entered into PSPP, an open-source SPSS alternative. Each participant was given a unique ID to note their row in the data set. This facilitated easier modification of any data that had been misinterpreted, omitted, or entered incorrectly. Using PSPP, the data was tested using ANOVA and Pearson's Correlation Matrix, and then the findings were collated and converted into useful data after analysis.

Analysis of variance (ANOVA) is used to determine whether there exists any statistically significant difference between three or more independent groups. Specifically, it tests for null hypotheses. The p-value denoted the statistical significance of the value between the groups. The closer the value is to 0, the greater the statistical value [26]. The updated data set consisted of 18 instances with responses in the set with the following attributes: student-student interaction (SS), student-content interaction (SC), student-teacher interaction (ST), age, gender, lab grade, lecture grade, and final grade. The status of the final results used the final GPA of the student interpreted as such: scores <2.00 were interpreted as a failure, which would mean that the student had to resit the module.

3.7 Limitations

There were a few limitations of the study. The researcher conducted the study on the groups. The use of convenience sampling may have had a part to play in the correlation, however failed to give a causal relationship in the study. It should be stemmed from the first limitation that although the invitation to participate was thrown out to 41 students in the groups, the response rate was approximately 46.3% with 19 responses. After validity checks were made, only 18 responses could be used.

4 Findings and Discussion

4.1 Descriptive Analysis

The total number of participants were 18 students. Sixteen students were males, and two were females. The students were all registered in the BSc in Animation Development and Production. The demographics can be seen in Table 2.

Table 2 Summary of instrument

Item	<i>N</i>	Measurement	Freq.	Approx. %
Gender	1	Male	16	88.8
	8	Female	2	11.1
Age	1	16–18	3	16.7
	8	19–21	8	44.4
		22–24	5	27.8
		25–27	2	11.1
Computer literacy level	1	Beginner	1	5.5
	8	Average	1	5.5
		Specialized	5	83.5
		Advanced	1	5.5
Faculty media awareness	1	Aware	9	50
	8	Unaware	9	50

4.2 Summary of Findings

Table 3 shows the academic performances for students using WhatsApp. The assessments, lab test, lecture test, individual assignments, and final grade were valued using the scores and GPAs, respectively. The participants in the control group had GPA scores at 2.57 and below, suggesting lack of participation as a prime factor. The lowest experimental group GPA score was reported as 2.00, while the highest GPA score was 3.71.

4.3 Research Questions

Research Question 1

What factors affected the relationship between social media interaction and academic performance?

Hypothesis 1 (H1): The three types of interactions would have a positive impact on the academic performance of the students, particularly ST on Facebook.

Hypothesis 0 (H0): The three types of interactions would not have a positive impact on the academic performance of the student.

With regard to the use of WhatsApp, there was a Pearson *r*-value of 0.19, as shown in Table 4, which suggested that there was a very weak but positive relationship between the student-content interaction and the final grade. The 2-tailed value of 0.444 heavily suggested that there was no statistical significance between the variables.

With regard to the use of WhatsApp, there was a Pearson *r*-value of 0.07, as seen in Table 5, indicating a very weak relationship existing between the variables

Table 3 Summary of the academic performance among the participant cases

Case	Control/ experimental	Lab test (%)	Lecture test (%)	Individual assignment (%)	Final grade (GPA)
A	E	65	62.5	62.5	2.00
B	E	62	50	34	2.00
C	C	87	56	84	1.43
D	E	87	62	62	3.14
E	E	80	64	81	3.42
F	C	58	50	0	0.00
G	E	85	64	74	3.42
H	C	50	43	80	2.29
I	E	90	90	98	3.71
J	E	60	80	82	2.57
H	E	80	50	67	2.86
K	C	0	0	0	0.00
L	E	87	90	82	3.71
M	C	40	40	53	1.71
N	C	40	40	27	1.14
O	E	80	20	62	2.00
P	E	100	70	62	2.57
Q	C	80	0	82	2.57

Table 4 Pearson’s correlation with averaged SC interaction and academic performance using WhatsApp

		Final grade
Student-content (SC)	Pearson’s correlation	0.19
	Sig. (2-tailed)	0.444

Table 5 Pearson’s correlation with averaged SS interaction and academic performance using WhatsApp

		Final grade
Student-student (SS)	Pearson correlation	0.07
	Sig. (2-tailed)	0.783

Table 6 Pearson’s correlation with averaged ST interaction and academic performance using WhatsApp

		Final grade
Student-teacher	Pearson correlation	0.42
	Sig. (2-tailed)	0.083

student-student interaction. The 2-tailed value of 0.783 tended toward 1 which suggests that there was no statistical significance. This suggested that there were no real conclusions to be drawn from the peer interactions on the WhatsApp platform.

With regard to the use of WhatsApp, there was a Pearson *r*-value of 0.42, as seen in Table 6, indicating a moderate relationship existing between the variable student-teacher interaction. The 2-tailed value of 0.083 tended toward 0 which suggests that there existed a statistical significance. This suggested that the more students communicated with their teacher in regard to the module on WhatsApp, the more possible it is that there would be a greater improvement of their academic performance.

Research Question 2

How would students feel about the use of the platform for learning and why?

This was a qualitative question with an optional response to which 16 students replied in an almost unanimously positive way. Some of the comments were:

It was really helpful being able to contact my instructor/classmates for help at anytime and keeping myself updated on important info regarding the course.

Very helpful. Firstly, feedback is almost always instant, which is especially helpful as opposed to emails. Material quickly gets passed along as well.

Very good and very convenient. Instructions, feedback, notice are a few of the most important things we need as a class body. WhatsApp played a major role in giving us these luxuries. Especially in a wide distance from instructors and classmates.

5 Conclusion

Much like the results in [5], there existed a weak but positive correlation between the variables SC and SS; however, there existed a moderately positive correlation with the ST variable which denotes that the most important interaction for the greatest academic success would be where students have direct contact with the teacher.

This study demonstrates that students are able to garner much more knowledge from the teacher and to a slightly lesser extent peers and content through the use of WhatsApp's nonintrusive social media setting. This increase in knowledge is due to information and class materials being readily available on demand even after class hours, which concurrently creates a more sociable environment to boost class morale and participation in the module.

Should faculty adopt this method of teaching, protocols will need to be established for information exchange and student/teacher conduct, so that the formality of the classroom setting isn't broken down.

For future studies, other collaborative nonintrusive means should be considered so that the adoption of new collaborative technology becomes a necessary staple in our ever-evolving culture especially in light of the global pandemic, COVID-19, which will shift education delivery to more virtual means in the short to medium term until a vaccine or cure is developed.

References

1. Z. Hussain, M. Rameez, R. Shah, N.A. Memon, WhatsApp usage frequency by university students: A case study of Sindh University. *Eng. Sci. Technol. Int. Res. J.* (52), 15–19 (2017)
2. D. Nitzza, Y. Roman, WhatsApp messaging: Achievements and success in academia. *Internet J. High. Educ.* 5(4), 255–261 (2016). <https://doi.org/10.5430/ijhe.v5n4p255>

3. M. Al-Mothana, M. Gasaymeh, University students' use of WhatsApp and their perceptions regarding its possible integration into their education. *Global J. Comp. Sci. Technol.* **17**(1) (2017) https://globaljournals.org/GJCST_Volume17/1-University-Students-use-of-Whatsapp.pdf
4. A.M. Elkaseh, K.W. Wong, C.C. Fung, Perceived ease of use and perceived usefulness of social media for e-learning in Libyan higher education: A structural equation modeling analysis. *Int. J. Inf. Educ. Technol.* **6**(3), 192–199 (2016) <http://www.ijet.org/vol6/683-JR159.pdf>
5. R.A. Jones, S. Bogle, An investigation of the use of Facebook groups as a learning management system to improve undergraduate performance. *World Congress Eng. Comput. Sci.* **1**, 211–216 (2017)
6. A.M. Gasaymeh, University students' use of WhatsApp and their perceptions regarding its possible integration into their education. *Global J. Comp. Sci. Technol. G Interdiscip.* **17**(1) (2017), Retrieved from https://globaljournals.org/GJCST_Volume17/1-University-Students-use-of-Whatsapp.pdf
7. World Bank, Jamaica, (2018), <https://data.worldbank.org/country/jamaica>
8. Business of Apps, WhatsApp revenue and usage statistics (Dec 2019). <http://www.businessofapps.com/data/whatsapp-statistics>
9. M. Irfan, S. Dhimmar, Impact of WhatsApp messenger on the university level students: A psychological study. *IJRAR* **6**(1), 572–586 (2019) https://www.researchgate.net/publication/332093442_Impact_of_WhatsApp_Messenger_on_the_University_Level_Students_A_Psychological_Study
10. R.S. Pol, S. Sathe, Ethical use of social media in student teacher communication with special reference to WhatsApp groups in Pune colleges. 1–11 (10 Jan 2019), Retrieved from <https://doi.org/10.2139/ssrn.3393897>
11. T.J. Ramabu, WhatsApp as part of a blended learning model to help programming novices. *World Acad. Sci. Eng. Technol. Int. J. Comput. Inf. Eng.* **13**(5), 250–254 (2019) Retrieved from <https://pdfs.semanticscholar.org/cd03/28f9763f5487ddd7654132e733eb402b9368.pdf>
12. F. Marikar, K. Alwis, S.N. Satharasinghe, D.A.P. Wickramasinghe, K.K.G.G.S. Kariyawasam, MOODLE's effectiveness in a developing country. *Online J. Dist. Educ. E-Learn.* **4**(3), 66–76 (2016)
13. H. Jamal, A. Shanaah, *The Role of Learning Management Systems in Educational Environments: An Exploratory Case Study (Masters in Informatics)* (Linnæus University, School of Computer Science, Physics and Mathematics, 2011), Retrieved from <http://nu.diva-portal.org/smash/get/diva2:435519/FULLTEXT01.pdf>
14. L. Darling-Hammond, K. Austin, S. Orcutt, D. Martin, Learning from others: Learning in a social context, in *The Learning Classroom: Theory Into Practice*, (2003), pp. 125–142, Retrieved from http://www.learner.org/courses/learningclassroom/support/07_learn_context.pdf
15. H. Meishar-Tal, G. Kurtz, E. Pieterse, Facebook groups as LMS: A case study. *Int. Rev. Res. Open Distrib. Learn.* **13**(4) (2012) Retrieved from <http://www.irrodl.org/index.php/irrodl/article/view/1294/2295>
16. A. Dutt, M.A. Ismail, Can we predict student learning performance from LMS data? A classification approach (2019), <https://doi.org/10.2991/iccie-18.2019.5>. Retrieved from https://www.researchgate.net/publication/334130244_Can_We_Predict_Student_Learning_Performance_from_LMS_Data_A_Classification_Approach
17. M. Gorhe, Impact of social media on academic performance of students (2019), <https://doi.org/10.13140/RG.2.2.21427.27687>
18. E. Lahuerta-Otero, R. Cordero-Gutiérrez, V. Izquierdo-Álvarez, Using social media to enhance learning and motivate students in the higher education classroom, in *Learning Technology for Education Challenges. LTEC 2019*, Communications in Computer and Information Science, ed. by L. Uden, D. Liberona, G. Sanchez, S. Rodríguez-González, vol. 1011, (Springer, Cham, 2019)

19. B. Urien, A. Erro-Garcés, A. Osca, *Educ. Inf. Technol.* **24**, 2585 (2019). <https://doi.org/10.1007/s10639-019-09876-5>
20. N.A. Zulkanain, S. Miskon, N.S. Abdullah, N.M. Ali, N. Ahmad, Communication and learning: social networking platforms for higher education, in *Emerging Trends in Intelligent Computing and Informatics. IRICT 2019*, Advances in Intelligent Systems and Computing, ed. by F. Saeed, F. Mohammed, N. Gazem, vol. 1073, (Springer, Cham, 2020)
21. T. Miyazoe, T. Anderson, The interaction equivalency theorem: Research potential and its application to teaching. *27th Annual Conference on Distance Teaching & Learning*, 1–6 (2011)
22. J.W. Creswell, A framework for design, in *Research Design: Qualitative, Quantitative, and Mixed Method Approaches*, 2nd edn., (University of Nebraska, Lincoln, 2003), p. 246: SAGE Publications Ltd. Retrieved from http://ucalgary.ca/paed/files/paed/2003_creswell_a-framework-for-design.pdf
23. O. Pilli, LMS vs. SNS: Can social networking sites act as a learning management systems? *Am. Int. J. Contemp. Res.* **4**(5), 90–97 (2014)
24. J. Rhode, *Interaction Equivalency in Self-Paced Online Learning Environments: An Exploration of Learning Preferences* (Capella University, 2008) Retrieved from <http://jasonrhode.com/pdfs/rhode-dissertation.pdf>
25. L.J. Cronbach, Coefficient alpha and the internal structure of tests. *Psychometrika* **16**, 297–334 (1951) Retrieved on April 23, 2017 from http://hbanaszak.mjr.uw.edu.pl/TempTxt/Cronbach_1951_Coefficient%20alpha%20and%20the%20internal%20structure%20of%20tests.pdf
26. Laerd Statistics, One-way ANOVA [research information] (2013), Retrieved from <https://statistics.laerd.com/statistical-guides/one-way-anova-statistical-guide.php>

Using Dear Data Project to Introduce Data Literacy and Information Literacy to Undergraduates



Vetria L. Byrd

1 Introduction

In the era of big data [24], and the ubiquitous nature of data, it is imperative that individuals have knowledge of basic data and information literacies. How and when should data literacy and information literacy be introduced? In the academic arena, students should be exposed to data and information literacy concepts early and often. Schuff [21] addresses the need for analytical thinking and information literacy as part of the foundational education of all students. Changes in scholarly landscapes, information delivery formats, and the evolving higher-education system necessitate innovative ways to engage students in preparation for effective lifelong learning skills that will impact their professional and personal lives. In order to equip students with the skills necessary to navigate the research landscape, and eventually their professional lives, libraries have engaged in information literacy training. Information literacy addresses how people learn and endows students with skills to “locate, evaluate, and effectively use information for any given need” [1]. While it is agreed, data literacy and information literacy are necessary educational building blocks, there are challenges associated with integrating data and information literacy into the undergraduate curriculum [12], and there is little consensus among educators on the best practices for integration. While the topics of data literacy and information literacy are deserving of dedicated courses, they tend to be folded under the umbrella of introductory data science courses. In this work, data literacy, information literacy, and visual communication are introduced to undergraduates enrolled in an introductory data visualization course.

V. L. Byrd (✉)
Purdue University, West Lafayette, IN, USA
e-mail: vbyrd@purdue.edu

This work aims to address the following research question: “What impact does the *Dear Data* [8] approach have on data literacy and information literacy skills of undergraduates in an introductory data visualization course?” The initial aim of the project was to create an assignment to reinforce concepts covered in class and motivate students to think differently about data and how that data is communicated. Elements of the *Dear Data Project* [9] were adapted as part of a 4-week assignment designed to assess whether or not the aims of this work were achieved. In the remaining sections, an overview of data literacy, information literacy, and the value of visual communication is presented, followed by a description of the course, and the modified *Dear Data Project*. Results from the student self-assessment of the skills targeted by the assignment are presented. The implications of the outcomes are discussed in the context of the research question previously posed, along with limitations and lessons learned. The paper concludes with recommendations for implementations of the assignment and integration of data and information literacy in future courses.

2 Background

There are a variety of approaches to teaching data literacy and information literacy. Historically, librarians have been the primary stewards of data, and they continue to be advocates of good data management practices [11]. A broad definition of data literacy is described as that which “enables individuals to access, interpret, critically assess, manage, handle and ethically use data” [19]. Matthews [17] provides four conceptual approaches to data literacy, each with a different focus of attention: research (an academic focus), classroom (a secondary education focus), carpentry (a practical training focus), and inclusion (a community development focus). Within secondary education, data literacy is promoted as a means for students to become familiar with data manipulation in problem-based learning [10]. The core learning objectives for a classroom focus in data literacy are question-asking, evidence gathering, and performing relatively simple analysis and visualization [17]. The work presented in this paper aligns with a classroom focus in data literacy with implications for research. In order to equip students with the skills needed to navigate the research landscape, and eventually their professional lives, libraries have engaged in information literacy training [22]. Information literacy addresses how people learn, and it gives students skills to “locate, evaluate, and effectively use information for any given need” [1]. The field of information literacy has become well established, and there is general agreement that students learn best when information literacy is integrated throughout the curriculum [20], but there is no consensus on how to implement the integration [18, 22, 26].

In 2016, Lupi and Posavic published the results of a year-long analog data drawing project entitled “Dear Data.” The project provided an interesting way of looking at and representing data. The “Dear Data” project has served as a catalyst for engaging students with data, from educational and social analysis [2] to simulating

the project to chronicle the process of becoming a researcher [6], for example. Lupi and Posavec [16] use data visualizations to communicate rather than the traditional written word. Each week, for 1 year, the authors chose a theme from everyday life to focus on requiring each author to figure out how to quantify and track their experiences and then visualize the resulting data set [8]. For the purpose of the research presented in this paper, the year-long *Dear Data* project was adapted to fit a 4-week assignment to reinforce content covered in class. Details of the adaptation approach are discussed below.

3 Participants

Participants of the project were students enrolled in CGT 270 Data Visualization course in fall 2019, their first data visualization course. The class met twice a week, for 50 minutes, over a 16-week semester. Students enrolled in the class had little or no experience visualizing data and no knowledge of the concepts of data and information literacy. A total of 36 students participated in the assignment. Twenty students were Computer Graphics Technology students majoring in data visualization; the remaining 16 students were non-majors. This created a unique opportunity to pair a data visualization major with a non-data visualization major, fostering near peer mentoring and network building among the pairs.

The non-data visualization majors were students participating in a campus-wide *Data Mine Learning Community Initiative* [7] at Purdue University. The *Data Mine* is designed to introduce undergraduates to data mining concepts and techniques from different academic perspectives. Students participating in the *Data Mine Data Visualization Learning Community* chose data visualization as their primary interest. Participation in the data mine is voluntary, but is also in addition to the course load for their academic majors. The *Data Mine* cohort consisted of students from a broad range of majors from computer science and engineering to business and aviation. The *Dear Data* Assignment served as a way to bridge the diversity of backgrounds and academic interests represented in the course and position everyone at the same starting point.

4 The Dear Data Postcard Visualization Assignment Methodology

The goal of the assignment is to introduce data literacy, information literacy, and visual communication as a first step in building data visualization capacity. Data drives visualization, so it is important that students have an understanding of what data is and how to represent and visually communicate the data. The Dear Data Assignment is intended to introduce these concepts in an engaging and active way.

The figure shows a postcard template divided into two main sections: the front (left) and the back (right).

Front Side (Left):

- Header: DEAR DATA, Week#: <Theme>, How to Read
- FROM: _____
- TO: _____
- Address: Data Buddy
- Instructions: *Use this space to provide details on how to read the postcard. This could include legends, icons, explanation for symbols, etc.*
- Warning: *This space can not be used to write an explanation for the visuals.*
- Graphic: A small square box containing a colorful starburst graphic.

Back Side (Right):

- Instruction: *Use this space to visually convey your ideas. The visual must align with the content provided in "How to Read" section on the other side.*
- Instruction: *You can and should use the entire space provided, but stay within the space designated as the postcard.*
- Instruction: *Remember, your data buddy will be viewing and interpreting the information you provide here.*

Fig. 1 Dear Data postcard template

The assignment was introduced as part of the first assignment in CGT 270 Data Visualization course. Prior to starting the assignment, students were introduced to the Dear Data Project [9] and shown examples of what a postcard looks like and the details of the original project. Unlike the original project, which took 12 months to complete, the duration of the course assignment was 4 weeks. The class assignment differed from the original Dear Data project in the following aspects: weekly class discussions about data and information literacy; students gained experience visually representing data from multiple sources (e.g., data they collect and data sets they are provided in the class); and the delivery of postcards via email. The logistics of the Dear Data Project was discussed in both sections. To simulate the essence of the 4-week Dear Data Assignment, students were randomly paired with a “Data Buddy,” a student enrolled in another section of the course. Before data buddies were assigned, each student was asked to complete a short survey indicating their interest in participating in the assignment, being paired with a data buddy from another section, and consenting to sharing their email address with their data buddy. Students were given the option to opt out of working with a data buddy from another section in lieu of being partnered with one of the course teaching assistants. Emails were sent, by the instructor, to each pair of data buddies to ensure each student facilitate their initial contact. Following the original *Dear Data Project* format, each week had a theme that was the same for each section. An example of the postcard template is shown in Fig. 1. The front and back sides of the postcard are shown side by side; however, a vertical layout is utilized in the paper postcard template used in the assignment. The left box shows the front of the postcard which contains recipient (To) and sender (From) information, the week number, theme for the week, and a “How to Read” section. Students were encouraged to create their own symbol or graphic where a postage stamp would be applied.

A new postcard template, with the theme for the week, was pre-printed and given to each student at the beginning of class each week for the duration of the assignment. Unless students indicated the need for using electronic devices for health reasons, postcards were created by hand, which means, students were expected to put pen/pencil to paper. Figure 1 shows instructions provided to students. Examples of completed postcards are provided in the Appendix. This assignment spanned the

first 4 weeks of the semester to coincide with the introduction of key concepts of data and information literacy. Each new theme aligned with the course topic for the week and was introduced at the beginning of the class. The last 30 minutes of the class was dedicated to allowing students to think about and start their postcards.

The content for each postcard should be reflective of the theme for the week and easily interpretable using the information students provided in the “How to Read” section of the postcard. Once the postcard is complete, students scanned the postcard with a mobile app or took a photo of the postcard and send it to their data buddy via email attachment. Students were instructed to save all of their postcards (sent and received) for the final part of the assignment.

The initial class session is dedicated to administrative tasks such as introductions, tour of the course webpage on Blackboard, assignments, and course goals, and objectives. The first Dear Data Assignment theme was “Introductions.” The postcard content could contain anything students wanted to share about themselves, but it had to be shared visually using meaningful hand-drawn characters and/or symbols rather than as short letters to their buddies.

The course topic for week 2 was data literacy. Data literacy was the ability to read, work with, analyze, and argue with data [17]. Much like literacy as a general concept, data literacy focuses on the competencies involved in working with data. It involves understanding what data mean, including the ability to read graphs and charts as well as draw conclusions from data [20]. Course topic and materials included, but was not limited to, understanding metadata, the difference between data generators and data consumers, and identifying appropriate data sources.

Students were instructed to be observant of their daily activities and think about what data looks like as they reflect on content covered in class on the topic. The theme and assignment for the second postcard was to visually represent “What Does Data Look Like?” based on randomly assigned training data sets exploring the data visualization process. The training data was acquired online from sample data on Tableau Public [25]. The Tableau website provides a buffet of data from diverse venues/categories (sports, public data, education, government, science, lifestyle, technology, health, entertainment, and business), in a variety of formats (.xlsx, csv, pdf, compressed files, json). The URL to the Tableau sample data was provided for students to download their datasets.

Information literacy was introduced in week 3. The theme for the third postcard was “Training Data.” Each students’ postcard visually described their randomly assigned Tableau Public training data (described above). Students were given the liberty to be as visually literal or abstract as they desired. A discussion on data types continued followed by an introduction to the first stages of the data visualization process: acquire, parse, and mine.

There is a notable trend toward storytelling with data [14], where a narrative is seen as the most effective way of getting across the lifecycle from goal to result and the new insights afforded by the analysis [13]. In week 4, students created a visual human portrait with data [23], using the postcards from weeks 1 to 3 as data sources. Students were asked to create a one-page infographic profile of their data buddy. For this part of the assignment, students were allowed to use any tools/software

they felt comfortable using to create the infographic. Students were not allowed to collect all of the postcards and paste them on a page and call it a profile. In week 4, instead of sending a postcard, each student sent the one-page infographic to their data buddy. At the end of the week, students were asked to create a “Dear Data Artifact.” The artifact consists of three sections: postcards received from their data buddy, postcards sent to their data buddy, and the one-page infographic profile they created of their data buddy. This artifact is a required component of each student’s course portfolio.

5 Postcard Assignment Assessment

Upon completion of the assignment, students were given a survey research instrument designed to assess the students’ perception of the assignment and, more importantly, assess if the instruction and expected learning goals for the class were achieved. Instructional goals for the assignment were to (1) develop the ability to draw reasonable inferences from data visualizations, (2) develop the ability to synthesize and integrate information and ideas, and (3) develop the ability to think creatively. The expected learning outcomes from the assignment are that students will (1) broaden their perceptions of what data is and how to visually represent it, (2) think more deeply about the different forms and types of data, and (3) think about how to simplify without sacrificing quality in representation of data.

The survey instrument consisted of ten statements that reflect the data literacy, information literacy, and visual communication skills targeted and utilized in the assignment. The survey instrument indicated level of agreement using a 5-point Likert scale (strongly agree, agree, neutral, disagree, and strongly disagree). Table 1 shows the categorization of the statements. The survey included two open-ended questions to allow students to include their opinion of what worked well (what they liked about the assignment) and what could be done differently or improved.

Table 1 Student self-assessment research survey instrument

Category	Statement: The Dear Data Assignment
Data literacy	S1. Helped me to understand what data means
	S2. Helped me to broaden my idea of what data is or could be
	S3. Helped me to think more critically about data
	S4. Improved my data literacy skills
Information literacy	S5. Helped me to see data in a variety of ways
	S6. Caused me to evaluate information more critically
	S7. Improved my information literacy skills
Visual communication	S8. Helped me to think critically about how to communicate my ideas without the use of technology
	S9. I was able to understand what my data buddy was trying to convey in the data postcards I received
	S10. Improved my visual communication skills

6 Results

7 Postcard Assignment Discussion

The goal of the assignment was to provide a hands-on experience using a proven, somewhat lost, mode of communication that challenges students to think about how to communicate visually and interpret messages conveyed on their data buddy's weekly postcards. Lastly, the assignment was intended to demonstrate new instructional methods for introducing data literacy, information literacy, and visual communication concepts. The assignment motivated students to take a step back from creating electronic documents for a moment and really think about how to effectively communicate visually and how to interpret visual communications.

Table 2 shows statistical indicators (mean, median, and mode) calculated to assess students' perception of the assignment's effectiveness in reaching the teaching and learning goals. Mean values greater than 3 are considered positive, and values less than three are considered negative. As seen in Table 2, the mean values for each statement are between 3.03 and 3.75, indicating, on average, students were more positive about the impact of the assignment. The results in Table 2 provide

Table 2 Statistical indicators for students' self-assessment of the assignment

Category of assessment	Stmnt#	Statement (<i>n</i> = 36)	Mean	Median	Mode
Data literacy	1	The Dear Data Assignment helped me to understand what data means	3.25	4	3.5
	2	The Dear Data Assignment helped me to broaden my idea of what data is or could be	3.56	4	4
	3	The Dear Data Assignment helped me to think more critically about data	3.28	4	3
	4	The Dear Data Assignment improved my data literacy skills	3.22	4	4
Information literacy	5	The Dear Data Assignment helped me to see data in a variety of ways	3.67	4	4
	6	Completing the Dear Data Assignment caused me to evaluate information more critically	3.03	4	3
	7	The Dear Data Assignment improved my information literacy skills	3.22	4	3
Visual communication	8	Helped me to think critically about how to communicate my ideas without the use of technology	3.75	4	4
	9	I was able to understand what my data buddy was trying to convey in the data postcards I received	3.75	4	4
	10	Q10. Improved my visual communication skills	3.70	4	4

insight into the research question, “What impact does the Dear Data approach have on data literacy and information literacy skills of undergraduates in an introductory data visualization course?” The results in Table 2 show students found in terms of data literacy the assignment helped to broaden their ideas of what data is or could be (Statement #2, mean = 3.56). In terms of information literacy, the assignment helped students to see data in a variety of ways (Statement #5, mean = 3.67) and in terms of visual communication (statements 8 and 9, mean values of 3.75) suggests students felt the assignment helped them to think critically about how to communicate their ideas, and they were able to interpret the data they received via postcards from their data buddy.

The simplicity of the assignment makes it ideal for the novice learning about data and information literacy, but it also presents elements of Bloom’s cognitive and metacognitive learning domains. The cognitive domain involves knowledge and the development of intellectual skills [3]. This includes the recall or recognition of specific facts, procedural patterns, and concepts that serve in the development of intellectual abilities and skills [15]. The metacognitive level includes the proper use, interpretation, discovery, inference, and the ability to evaluate and create content. All of these elements are encapsulated in the assignment and presented in an engaging way that allows students to demonstrate what they know, through visual representation and communication. Such explaining involves the kind of capacities labeled “analysis” and “synthesis” in Bloom’s Taxonomy [3]. Students with in-depth understanding in this sense have greater control over data and over robust connections – than those with limited understanding.

7.1 What Does the Assignment Teach?

This assignment was used to introduce data literacy and information literacy concepts to an undergraduate class of data visualization majors and non-majors. The “Dear Data” assignment harks back to a more nostalgic era, unbeknownst to many students, when there was deliberation over information taken in and offered to others [23]. “To draw is to remember” [16]. This assignment is intended to introduce new concepts of data and information literacy to aid students in visually communicating their ideas. Students enrolled in the course from various backgrounds, interests, and academic preparation. The Dear Data Assignment, from an instructional perspective, enables the establishment of a baseline for class content to ensure all students begin with basic understanding of literacies associated with producing and interpreting data. The assignment introduces students to the process of visualizing data in an active way. This work contributes to the knowledge base of data visualization pedagogy and data visualization capacity building. Upon completing the introductory data visualization course, students will demonstrate their knowledge of the data visualization process and, more importantly, understand the data and information driving the visualization.

7.2 *Limitations*

Fall 2019 was the first semester this assignment was offered as part of the course. There is room for refinement and improvement in logistics and content. The approach is not devoid of limitation, some of what are inherent in the “Dear Data” project. Some students did not like the idea of not being able to use electronic devices to create postcards; they found the paper-based method to be archaic. While their points were valid, the use of electronic devices to create the postcard defeated the purpose of the assignment: to focus more closely on content rather than electric devices to enhance visual output. Some students felt they lacked drawing and sketching skills needed to convey their thoughts visually. To address this issue, students were reassured their postcard would be assessed based on the insight conveyed and reminded of the purpose of the “How to Read” section of the postcard. The need for a “How to Read” section is a challenge that is inherent in the original Dear Data Project [23]. Students indicated they would have liked to have had more flexibility regarding themes for the postcards. Unsure of the outcome of the assignment, the selection of weekly themes was strategically rigid for consistency within and across sections.

There are plans to offer the assignment again, leveraging lessons learned. Data buddy pairs will be given the opportunity to choose their theme for the week, but the theme must be related to topics covered in class. The duration of the assignment will be revisited to determine if the assignment should be a semester-long assignment. There were a couple of instances where students either joined the class after the assignment started or dropped the class before the assignment concluded, creating a scenario where the remaining student(s) is without a data buddy. Each instance was handled individually. In this scenario, there were two such cases, and they were solved by utilizing course teaching assistants as data buddies. This could be challenging if more students joined late or dropped the class in the middle of the assignment. In those cases, the students with absent data buddies would serve as data buddies to each other. The second logistical challenge was students in one section forgetting to send their postcard to their data buddy in the other section. Each week, students were asked to confirm (verbally) if they had received a postcard from their data buddy. Students who did not send a postcard were sent a reminder from one of the teaching assistants that their postcard was pending. This usually resulted in the delivery of the postcard.

8 **Conclusion**

In this work, a hands-on method for introducing data literacy and information literacy was presented. The assignment is designed to help students collect and generate data, develop the ability to draw reasonable inferences from data, develop a practice of synthesizing and integrating information and ideas, and develop the practice of thinking creatively and critically about data.

We acknowledge the study of data literacy, information literacy, and visual communication requires more than 4 weeks of study and implementation with more rigor than creating postcards; however, the results support the argument that this assignment is suited for students new to data literacy and information literacy. Data and information literacy are fundamental to understanding and implementing the data visualization process. These literacies are strategically covered at the beginning of CGT 270 Data Visualization course [4] to prepare students for a more in-depth mapping and application of learning taxonomies to support building data visualization capacity [5].

Future implementations of the assignment will include the development of a rubric to evaluate postcard content as to the assignment’s effectiveness of reaching the research aims and a quasi-experimental design to compare the outcomes and perceptions of data visualization majors and non-majors.

A.1 Appendix

In this section are shown samples of postcards created for the assignment. Figure 2 shows a pair of postcards created and shared between a data buddy pair introducing

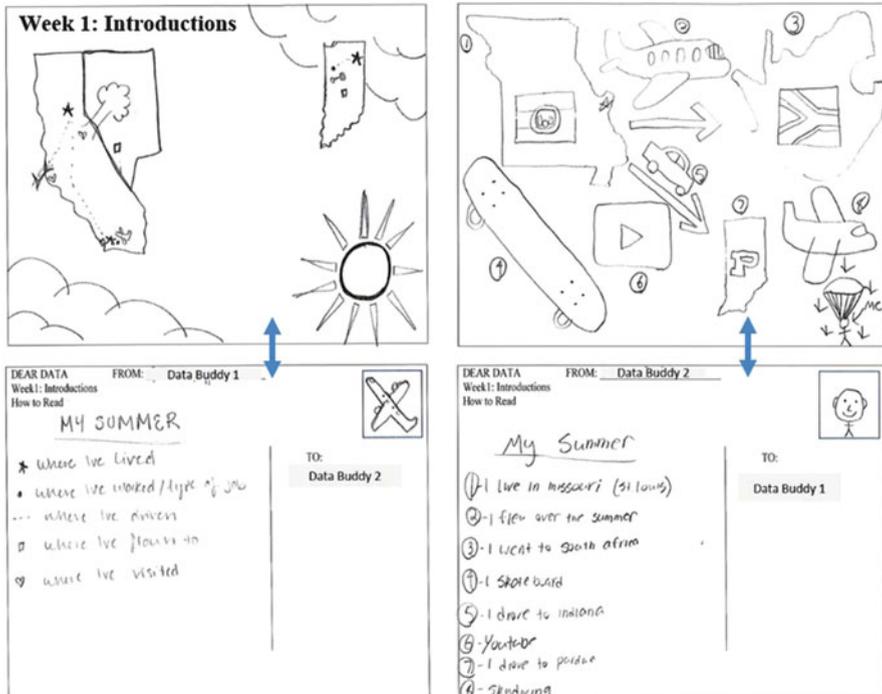


Fig. 2 Dear Data postcards from week 1: introductions

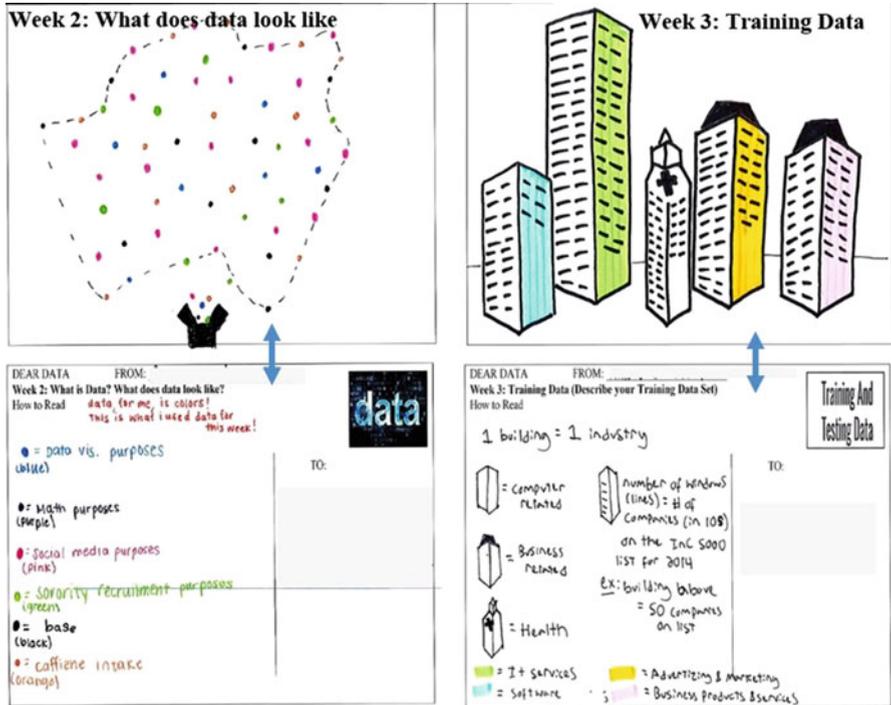


Fig. 3 Dear Data postcards from week 2 (what does data look like) and week 3 (training data)

themselves. Figure 3 shows a postcard from week 2 and week 3, with the “What does data look like?” and “Training Data” themes, respectively.

References

1. Association of College and Research Libraries, Presidential Committee on Information Literacy: Final report (1989), <http://www.ala.org/acrl/publications/whitepapers/presidential>
2. C. Badenhorst, B. FitzPatrick, Emerging researcher pedagogies: The “Dear Data” project. *The Morning Watch: Educational and Social Analysis* 45(3–4 Winter) (2018)
3. B.S. Bloom, Bloom’s taxonomy (1956), https://en.wikipedia.org/wiki/Bloom%27s_taxonomy.
4. V. Byrd, Introducing data visualization: A hands-on approach for undergraduates, in *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, (Association for the Advancement of Computing in Education (AACE), 2018), pp. 730–736
5. V. Byrd, Using Bloom’s taxonomy to support data visualization capacity skills, in *E-Learn: World Conference on E-Learning in Corporate, Government, Healthcare, and Higher Education*, (Association for the Advancement of Computing in Education (AACE), 2019), pp. 1039–1053

6. C. Cumby, Dear Data: The process of becoming a researcher. *The Morning Watch: Educational and Social Analysis* **45**(3–4 Winter) (2018)
7. Data Mine, Purdue University (2019), <https://datamine.purdue.edu/>. Accessed 03/24/2020
8. Dear Data, Publishers Weekly **263**(28), 57 (2016)
9. Dear Data Project (2016), URL <http://www.dear-data.com/> Last accessed 03/24/2020
10. R. Erwin, Data literacy: Real-world learning through problem-solving with data sets. *Am. Second. Educ.* **43**(2), 18–26 (2015) Retrieved from <http://search.ebscohost.com/login.aspx?direct=true&db=eue&AN=103298736&site=ehost-live>
11. K. Fonticiaro, J.A. Oehrli, Why data literacy matters. *ALA Knowl. Quest* **44**(5), 20–28 (2016)
12. K. Hunt, The challenges of integrating data literacy into the curriculum in an undergraduate institution. *IASSIST Q.* **28**(2–3), 12–12 (2005)
13. IODC, Enabling the data revolution (2015), Retrieved from <http://1a9vrva76sx19qtvg1ddvt6f.wpengine.netdna-cdn.com/wp-content/uploads/2015/11/opendatacon-report-en-web.pdf>. Accessed 03/24/20
14. R. Kosara, J. Mackinlay, Storytelling: The next step for visualization. *Computer* **46**(5), 44–50 (2013)
15. D.R. Krathwohl, L.W. Anderson, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives* (Longman, 2009)
16. G. Lupi, S. Posavec, *Dear Data* (Chronicle Books, 2016)
17. P. Matthews, Data literacy conceptions, community capabilities. *J. Community Inform.* **12**(3), 47–56 (2016)
18. R. Monge, E. Friscaro-Pawlowski, Redefining information literacy to prepare students for the 21st century workforce. *Innov. High. Educ.* **39**(1), 59–73 (2014)
19. J. Prado, M. Marzal, Incorporating data literacy into information literacy programs: Core competencies and contents. *Libri* **63**(2), 123–134 (2013). <https://doi.org/10.1515/libri-2013-0010>
20. I.F. Rockman, *Integrating information literacy into the higher education curriculum: practical models for transformation*, 1st edn. (Jossey-Bass, San Francisco, 2004)
21. D. Schuff, Data science for all: A university-wide course in data literacy, in *Analytics and Data Science: Advances in Research and Pedagogy*, Springer Annals of Information Systems Series (<http://www.springer.com/series/7573>), ed. by A. Deokar, A. L. Gupta Iyer, M. C. Jones, vol. 21, (2017), 2016–2017
22. Y. Shorish, Data information literacy and undergraduates: A critical competency. *Coll. Undergrad. Lib.* **22**(1), 97–106 (2015)
23. D. Silverberg, The minute data of everyday life on 52 postcards over 52 weeks. Washingtonpost.com, Washingtonpost.com, 26 Aug 2016
24. I.Y. Song, Y. Zhu, Big data and data science: What should we teach? *Expert. Syst.* **33**(4), 364–373 (2016)
25. Tableau, Public data sets (2019), https://public.tableau.com/s/resources?qt-overview_resources=1
26. R.I. Waller, G. Miller, D.M. Schultz, Information literacy: Benefits, challenges and practical strategies, in *Handbook for Teaching and Learning in Geography*, (Edward Elgar Publishing, 2019)

An Educational Tool for Exploring the Pumping Lemma Property for Regular Languages



Josue N. Rivera and Haiping Xu

1 Introduction

The regular languages and finite automata are some of the most studied topics in formal language theories [1]. The notion of finite automata, introduced by McCulloch and Pitts in 1943, revolutionized the idea of what a computational model looks like, which has brought significant contributions in computer science and engineering [2]. These include but not limited to the ideas of perceptrons (predecessors to neural networks) and logic design used in the development of modern embedded systems [3]. The significant impact that finite automaton and regular languages had made in modern civilization is well documented.

Despite thorough studies and many existing educational materials for regular languages and finite automata, the pumping lemma for regular languages has been a very difficult topic for students to understand in a theoretical computer science course. Due to a lack of tool support, students usually have insufficient practice to clearly understand the concept of pumping length and how to prove a language is not regular using the pumping lemma. In this paper, we introduce an active learning tool called MInimum PUmping length (MIPU) educational software to explore the pumping lemma property for regular languages. The goal of MIPU is to serve as an active learning tool for students to understand the pumping lemma

This material is based upon a project for honored course CIS 560: Theoretical Computer Science, University of Massachusetts Dartmouth.

J. N. Rivera · H. Xu (✉)

Computer and Information Science Department, University of Massachusetts Dartmouth,
Dartmouth, MA, USA

e-mail: josue.n.rivera@umassd.edu; hxu@umassd.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_11

143

property, which is an essential concept revealing the relationship between regular languages and finite automata through its formal proof. Active learning has been defined as a high-level learning process where students are the primary actors in the process [4]. Unlike the traditional learning model where students learn new concepts through a medium such as a textbook, active learning requires students to perform hands-on tasks and learn by doing. The aim of active learning is to have students learn from experience instead of being informed about the ideas with little practical engagement. Hence, in recent years, active learning models have become a focus of discussion for teaching students in the classroom. They have been found to be effective in enhancing students' retention, boosting higher-order thinking and reasoning skills, and improving student performance in STEM courses [5].

As an intriguing property of regular languages, the pumping lemma allows one to prove a language is not regular by showing the language does not satisfy the pumping lemma property. Such a proof requires one to clearly understand the concept of pumping length, how a string can be split into substrings in accordance with the property, and how it can be pumped. With MIPU, we attempt to provide three major features that contribute to the overall understanding of the pumping lemma and the concept of minimum pumping length. First, the software assists in verifying if a string belongs to a regular language described by a regular expression. By converting a regular expression into a finite automaton, we can determine if a string is a member of a given regular language. Second, the software can generate a list of short strings of a regular language. As a regular expression defines the pattern of a regular language, by generating the short strings, students can gain a better understanding of the language. Lastly, this tool can automatically calculate the minimum pumping length of a regular language and demonstrate how a given string belonging to the regular language can be split into three substrings that satisfy the pumping lemma property.

2 Related Work

With the advance of powerful personal computers and the Internet, access to educational tools become much closer within reach than in any other time in history. Gradually, educational tools have become widely available online that help to explain many advanced topics in a variety of fields. With the rise, there has been an increasing number of active learning applications that focus on aiding STEM education – a critical subject to teach in our modern lives. Computer science education is particularly crucial due to the numerous influential advancements that have emerged from the field. Thus, no wonder that many of these applications introduced are directed toward enhancing the experience of learning complex topics in the areas of study.

It has been proven that active learning can strengthen the experience of STEM students in the classroom [5]. A research performed by Kim and her colleagues

in 2012 elaborated on and described the effects that active learning modules may have in enhancing students' critical thinking [6]. Their study had two goals: to examine the levels of critical thinking exhibited in individual reports over the semester and to explore the effect of active learning on undergraduate students' critical thinking. With the goals in mind, they focused on designing appropriate strategies to foster innovation in an undergraduate general science course. Their team used the strategies to support students in engaging in hands-on practice by providing the learning environments that required the use of scientific knowledge in solving real-life problems. The designs included support of cognitive process such as scaffolding strategies and tools for building a knowledge pool. The modules presented to the students to evaluate critical thinking dealt with the understanding of evacuation plans for hurricanes and authentic problems associated with global warming. The study showed that the active learning strategies had been helpful to promote students' critical thinking. In recent years, there has been a push to bring effective active learning tools and strategies into the classroom to enhance the learning process of students. This trend has greatly motivated our research in developing effective tools to support active learning in computer science.

The use of educational tools in computer science classrooms has seen a significant emergence. Computer science is now an integral part in the society that we live in for the role that it plays in many crucial aspects of it. In a recent paper, Wang from the University of Toledo tackled the integration of educational tools in computer science courses [7]. He presented multiple modern software tools to assist with various subjects in a database course. He first introduced different components in a typical database course, such as Entity Relation (ER) diagram and MySQL. Then he introduced existing support tools that make the various component more interactive and easier to learn. The result of implementing these strategies in his online database course was an increase in the visual appeal of the taught contents along with a significant jump in the average grade of the class in various subjects. While his research was intended to be applied to online courses, the principles learned can be easily transferred to an in-person setting. Wang's work is an example of the shift in computer science education that is attempting to make learning more interactive and enable topics to be learned from experience rather than through passive learning.

There are currently many existing tools for experimenting with topics related to formal languages and automata, such as deterministic finite automata (DFA), nondeterministic finite automata (NFA), conversion from NFA to DFA, pushdown automata (PDA), and multi-tape Turing machines. Among the existing tools, the Java Formal Languages and Automata Package (JFLAP) is by far one of the most popular educational tools. JFLAP is a collection of graphical tools that can be used as an aid in learning the basic concepts of formal languages and automata theory [8] [9]. The goal of the tool is to "enhance the formal languages course, changing it from a traditional mathematics course into a 'hands-on' computer science course" [10]. In JFLAP, the graphical interface allows one to build automata, run them with different input strings, and see a snapshot of the automaton at any stage of the computation along with the different configurations that lead to a final state. Despite being a powerful tool, JFLAP lacks in some major areas of formal languages and finite

automata theory, e.g., the tool support for calculation of minimum pumping length and facilitating students to understand pumping lemma property. To the best of our knowledge, there are no existing educational tools that support those features. As such, our work is complementary to other research efforts, e.g., JFLAP, that use software tools to support hands-on computer science education.

3 Tool Support for Pumping Lemma

Pumping lemma is a theoretical idea that cannot be easily presented to students through a traditional visual medium or an intuitive explanation. Instead, it requires students to go through a sufficient number of cases to build a mental model of the concepts. Therefore, the design of an effective active learning tool for understanding pumping lemma is crucial for a successful education in theoretical computer science.

3.1 *Pumping Lemma for Regular Languages*

Aiding students in understanding pumping lemma is the core goal of MIPU. Pumping lemma is a property that all regular languages have, which can be demonstrated using a finite automaton. For this reason, it is important to understand finite automata to learn how pumping lemma works. A regular language is defined as a set of strings that can be accepted by some finite automaton. A finite automaton is commonly seen as a computational model with a limited number of states that contain transitions between states labeled by symbols from a finite alphabet. Some or none of the states in a finite automaton are accept states and one of the states is a start state. To compute an input string, an automaton reads each symbol in the string in order and transitions to states according to the transition function. Once all symbols in the string have been processed, if a current state of the automaton is an accept state, the string is accepted; otherwise, the string is rejected. Two types of finite automata are DFA and NFA, which are equivalent. The strength of finite automata emerges from its ability to represent real-world computation using a simple model. The act of switching on and off a light is one such example, but finite automata can be used to model more complicated situations, e.g., representing the states of characters in a game or performing pattern recognition on strings.

An intuitive way of distinguishing regular languages from non-regular languages is to determine if the modeling machine needs to have an unbounded memory to account for the unlimited number of possibilities. However, this intuitive approach does not always work. For example, in the following two languages C and D , both are seemingly non-regular, but surprisingly, one of them (language D) is in fact regular [11].

$$C = \{w \mid w \text{ has an equal number of 0s and 1s}\}.$$
$$D = \{w \mid w \text{ has an equal number of occurrences of 01 and 10 as substrings}\}.$$

We can formally prove language D is regular by designing a regular expression that describes the language. However, one may try to design a regular expression to describe language C , but still fail to find one. Can we conclude C is not regular because no one is able to design a regular expression to describe C ? The answer is no, and thus, it is important to establish a formal approach to assisting in determining the non-regularity of a language.

The pumping lemma for regular languages is a technique for proving non-regularity. The pumping lemma states that all regular languages have a special property, i.e., the pumping lemma property. Therefore, if a language does not demonstrate the pumping lemma property, the language must be non-regular. The pumping lemma ensures that any string in a regular language with at least a certain length, i.e., the pumping length p , can be “pumped” and still belong to the language. Pumping a string, in the context of the property, refers to repeating or eliminating a section of a string and still maintaining its membership with the language.

The pumping lemma can be described as follows [11]: if A is a regular language, then there is a positive number p (the pumping length) where if s is any string in A with a length of at least p , then s can be divided into three substrings, $s = xyz$, satisfying the following three conditions:

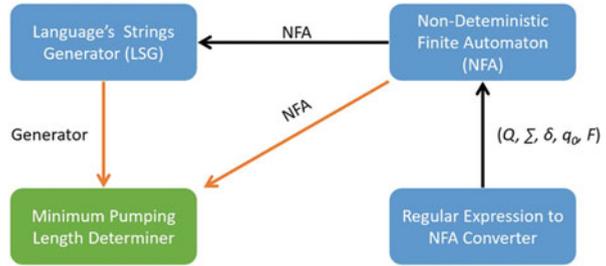
1. For each $i \geq 0$, xy^iz belongs to A .
2. $|y| > 0$.
3. $|xy| \leq p$.

As demonstrated earlier, intuitively understanding the regularity and non-regularity of a language might not be sufficient. Pumping lemma has played an important role in helping to understand regularity and proving a language is not regular by contradiction. However, a correct proof for non-regularity of a language requires accurate understanding of the pumping lemma for regular languages. The goal of MIPU is to aid in understanding the pumping lemma property, and based on the conditions required to satisfy the pumping lemma property, the tool provides three major functionalities: membership testing, generation of strings that belong to a regular language, and calculation of the minimum pumping length needed to demonstrate the existence of the property in a language if it is regular.

3.2 A Framework of the Active Learning Tool

To make MIPU easily customizable and flexible to optimize, it was built with an object-oriented design (OOD) in mind. This would enable specific components of the tool to be adjusted without affecting the overall functionality. The framework of MIPU consists of four major components that represent the major concepts in formal languages and automata. Figure 1 showcases their corresponding classes and their interactions with each other.

Fig. 1 A framework of MIPU with four major components



As shown in Fig. 1, the four components of MIPU are a regular expression to NFA converter, an NFA simulator, a language's strings generator (LSG), and a minimum pumping length determiner. For membership testing, the regular expression to NFA converter is used to transform a given regular expression into an NFA instance that can be easily operated on. This NFA instance is bundled with a “compute” function that is used to determine if a given string is a member of the language. To generate short strings, the language's strings generator is used to generate a list of such strings that belong to the language described by the regular expression. Lastly, the determination of the minimum pumping length of a regular language uses all the components in MIPU as needed by the pumping lemma for regular language. These functionalities are further discussed in Sect. 4.

Regular Expression to NFA Converter

The regular expression to NFA converter takes a regular expression in the form of a string and decodes it into a tuple of five elements that comprise an NFA. These elements include a finite set of states (Q), a finite set of the alphabet that forms the language (Σ), the transition function between states (δ), a start state (q_0), and, finally, a set of accept states (F). Algorithm 1 shows how to generate these elements of 5-tuple. The algorithm first checks if the regular expression represents a base case, which can be an empty set, an empty string, or a regular expression containing only one symbol. Then the regular expression is parsed into a list of segments that can be iterated through to form an NFA.

Algorithm 1 Convert a regular expression into an NFA

Input: regular expression $regExp$

Output: T as 5-tuple ($states$, $alpha$, $transfun$, $startq$, $acceptq$)

- 1: initialize $states$ and $alpha$ to empty sets
- 2: initialize $transfun$ to an empty map with state and symbol as key and traversable states as value
- 3: $currq = 0$
- 4: $createNFA(regExp)$
- 5: **if** $regExp$ is an empty set

```

6:   return  $T$  with  $q_{currq}$  as start state and no accept state
7:   else if  $regExp$  is the empty string
8:   return  $T$  with  $q_{currq}$  as the start and accept state
9:   else if  $regExp$  is of length 1
10:    add transition between  $q_{(currq++)}$  and  $q_{(currq++)}$  with  $regExp$  as the
transition symbol
11:    add  $q_{(currq-1)}$  and  $q_{(currq-2)}$  to the states set
12:    add  $regExp$  to the alphabet set
13:    return  $T$  with  $q_{(currq-2)}$  and  $q_{(currq-1)}$  as the start and accept state,
respectively
14:     $seg = parseSegments(regExp)$ 
15:    for each segment  $s$  in  $seg$ , where  $s$  is not an operation
16:       $T_{seg} = createNFA(s)$ 
17:       $start\_seg[s] =$  start state of  $T_{seg}$ 
18:       $accept\_seg[s] =$  accept state of  $T_{seg}$ 
19:      for each segment  $s$  in  $seg$ , where  $s$  is star
20:        update  $currq$  and add new states to  $states$  set
21:        add transitions starting with start state of the previous
segment and ending with  $q_{(currq+2)}$ 
22:      for each segment  $s$  in  $seg$ , where  $s$  is concatenation
23:        update  $currq$  and start & accept states
24:        add epsilon transition between the previous segment
and the next segment
25:      for each segment  $s$  in  $seg$ , where  $s$  is union
26:        update  $currq$  and add new states to  $states$  set
27:        add transitions to connect the previous segment and
the next segment
28:    return  $T$  with start and accept state of  $seg$ 

```

For the symbol that represents the empty set, an NFA is returned with the current state ($currq$) as the start state, and there is no accept state. For the empty string, an NFA is returned with $currq$ as both the start and accept state. Lastly, for a regular expression that contains only one symbol other than a regular operation, two states are created ($currq++$ and $currq++$), which are connected by a transition labeled by the symbol. When the regular expression does not represent a base case, it is parsed into a list of segments. The procedure utilized to parse the expression into segments will later be discussed in Algorithm 2. The segments are iterated through in four different *for*-loops. The first *for*-loop traverses all the elements that are not an operation and perform recursive calls on Algorithm 1 for the individual segments until the base cases are reached. The following three *for*-loops are ordered according to the precedence of the regular operations, namely, star, union, and concatenation. For each regular operation, the algorithm follows the standard regular expression to NFA conversion techniques [11]. New states and transitions are added as needed to the segment(s) that the operation is applied to; meanwhile, $currq$ is also updated. The start and accept state of the segments involved synchs to reflect in the newly

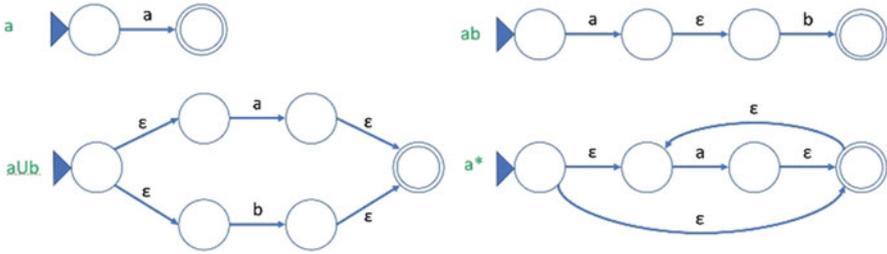


Fig. 2 Conversion of regular expression to an NFA

created NFA. It must be highlighted that for union and star operations, the NFA is adjusted to contain a single accept state. Figure 2 showcases these changes. After all the segments are constructed, the 5-tuple representing an NFA is returned.

To make the conversion procedure from a regular expression into an NFA more flexible and efficient, Algorithm 2 is used to section a regular expression into segments while building the entire NFA.

Algorithm 2 Parse a regular expression into a segment list

Input: regular expression *regExp*
Output: expression segment list *seg*

- 1: `parseSegments(regExp)`
- 2: initialize *count* to 0 and *temp* to an empty string
- 3: initialize *seg* to an empty list of strings
- 4: **for** $i = 1$ **to** $regExp.length$
- 5: **if** $regExp.charAt(i) == '('$
- 6: $count++$
- 7: **if** $count == 1$ **continue**
- 8: **else if** $regExp.charAt(i) == ')'$
- 9: $count--$
- 10: **if** $count == 0$
- 11: add *temp* to *seg* and reset *temp* to an empty string
- 12: **if** $i < regExp.length - 1$
- 13: add concatenation operation “.” to *seg* if needed
- 14: **continue**
- 15: $temp += regExp.charAt(i)$
- 16: **if** $count == 0$ // *temp* is an operation or one symbol
- 17: add *temp* to *seg* and reset *temp* to an empty string
- 18: **if** $i < regExp.length - 1$
- 19: add concatenation operation “.” to *seg* if needed
- 20: **return** *seg*

Algorithm 2’s role is to decipher a regular expression into a list of segments that Algorithm 1 can easily convert into an NFA. The algorithm traverses each symbol

of the regular expression while, at the same time, it keeps track of the appearance of parenthesis (*count*), the segments of the expression (*seg*), and a temporary buffer for the current segment (*temp*). For each character iterated, the character is first processed to discern parentheses. This step is performed to determine if the upcoming elements of the expression are isolated from the rest of the elements. This is essential for operations like union that requires all the elements to the right and left of the operation to be passed as inputs. If the current character is an opening parenthesis, *count* is increased by one, and the procedure immediately moves on to the next symbol. On the other hand, if the character is a closing parenthesis, *count* is decreased by one, and the collected elements in *temp* is added into *seg* when *count* becomes zero. In addition to the elements added thus far, a concatenation operation is added as well if the next character is not a *star* or *union* operation. These components ensure that isolation is secured. If the character is not a parenthesis, it is added into *temp*. When *count* equals zero, *temp* must contain an operation or a single symbol, which is added into *seg*. In this case, a concatenation operation is added if needed. To better illustrate the functionality of Algorithm 2, a sample input and its corresponding output are provided as follows:

Input = “a(caUac)c*cac”.

Output = [“a”, “.”, “caUac”, “.”, “c”, “*”, “.”, “c”, “.”, “a”, “.”, “c”].

One aspect of Algorithms 1 and 2 that must be highlighted is that they require the omission of special characters as element in the NFA alphabet. The character used to represent union, concatenation, star, empty language, and epsilon cannot be elements in the alphabet. Due to this notion, the algorithms have default characters that they treat as these special symbols. Union is represented by uppercase letter “U”; concatenation is portrayed by the period “.”; and the star operation is symbolized by the star character “*”. The empty language is equivalent to the backslash (\), and lastly, the empty string epsilon is depicted by lowercase letter “e”. Future improvement to MIPU will allow customized settings to overwrite the default characters used.

Nondeterministic Finite Automaton (NFA)

The NFA class in the framework takes the 5-tuple generated by the regular expression to the NFA converter and offers methods for managing the NFA. One such method is to test membership of an input string. To compute the input string, the states of the NFA are traversed based on the symbols in the input string, and membership is determined if one of the possible paths leads to an accept state. This NFA model is passed to the language’s strings generator and the minimum pumping length determiner for each to serve their respective roles.

The membership testing of an input string results from three individual algorithms that contribute to each other to decide if the current state ends is an accept state after a string is computed. Algorithm 3 shows this process that iterates through the character in an input string and transits to other states based on the character

read. At the end of the iteration, this algorithm returns true or false depending on whether or not the current is found to be an accept state.

Algorithm 4 performs the transition method used in Algorithm 3. The algorithm searches for all possible states that the current list of states can traverse to. It will then remove those states and update the list to reflect the most recent version of the states that the current list of states has moved to. As the NFA may have multiples states that it can traverse to from the current state and an input symbol, the transit algorithm (Algorithm 4) is separated from Algorithm 3 for simplicity.

An intriguing property of the NFA is the use of a special transition called *epsilon* transition. An epsilon transition allows for the finite automaton to traverse without the need of an input symbol. The traversal of this type of transition is encapsulated in Algorithm 5. The algorithm iterates a changing list that updates within the method itself. The logic behind this approach is that if an epsilon transition is found, it is possible that the destination state may also contain another epsilon transition leading to another state. However, this method has a hidden issue: if a cycle of epsilon transitions exists, this would lead to an infinite loop. The solution to this is to check if a new traversed state already exists in the list before it is added into the current list.

Algorithm 3 Compute a string

Input: *inputStr*, *transitions*

Output: *membershipStatus*

- 1: initialize *current* to an empty list
- 2: add start state to *current*
- 3: `updateEpsilonTransitions(current, transitions)` // Algorithm 5
- 4: **for each** symbol *c* in *inputStr*
- 5: `transitState(c, current, transitions)` // Algorithm 4
- 6: `updateEpsilonTransitions(current, transitions)` // Algorithm 5
- 8: **if** *current* state is an accept state
- 9: **return** true
- 10: **else**
- 11: **return** false

Algorithm 4 Transit between NFA states

Input: *symbol*, *current*, *transitions*

Output: *current*

- 1: `transitState(symbol, current)`
- 2: **if** *symbol* is epsilon
- 3: **return** *current*
- 4: *size* = the size of the *current* list
- 5: **for** *i* = 1 **to** *size*
- 6: **if** there is a *transition* for current state *i* and *symbol*
- 8: **for each** traversable state *s* from *current* state *i*

```

9:         if state s is not a member of current
10:            add state s to current
11:    remove state i from current
12:    return current

```

Algorithm 5 Update epsilon transitions

Input: *current*, *transitions*

Output: *current*

```

1: updateEpsilonTransitions(current, transitions)
2:   for each state i in current // current changes in the loop
3:     if there is an epsilon transition from current state i
4:       for each traversable state s from current state i
5:         if state s is not a member of current
6:           add state s to current
7:   return current

```

The algorithms presented form the bases for the membership testing functionality of MIPU. After traversing the NFA graph and tracking all possible paths, one can determine the membership of a string by observing if one of the paths leads to an accept state. The ability to detect the membership of a string is essential for the next two components of the MIPU framework, namely, the language's strings generator and minimum pumping length determiner.

Language's Strings Generator (LSG)

The language's strings generator uses a given NFA instance to generate an adjustable number of permutations from the alphabet. These permutations must be strings that can be accepted by the finite automaton. Every so often, the generator generates a new batch of strings and stores them in a buffer for future usage. To improve the performance of the permutation process for strings, branches of a permutation tree are tracked. If a path will not likely lead to a final state along the way, that branch is removed. The fate of a future branch can be determined by observing the current states that the NFA is tracking for the current segment of the string that has been generated thus far.

Minimum Pumping Length Determiner

Finally, as one of the primary functionalities of MIPU, the minimum pumping length determiner can calculate the minimum pumping length of a regular language according to the definition of pumping lemma. The tool also retrieves one of the shortest strings in the language that meet the conditions and partitions it into three

segments x , y , and z described in pumping lemma. The method takes an NFA instance and the strings generated by the LSG as inputs and tests the conditions to derive how the pumping lemma property is satisfied. Since the strings are ordered by their string lengths, we will be able to check strings starting from the shortest one and determine the minimum pumping length that meets the pumping lemma requirements.

4 Pumping Lemma for Regular Language

The pumping lemma presents a set of conditions that must be satisfied in order to demonstrate the pumping lemma property. These conditions include testing the membership of a “pumped” string, where the original string belongs to a regular language and is of a size greater than or equal to the minimum pumping length. To help with the correct understanding of the pumping lemma concept, MIPU offers three main tools that are essential to determine the existence of the property in regular language, which are membership testing, string generation, and automated minimum pumping length determination, as illustrated in Fig. 3. Membership testing function determines if an input string is a member of a given regular language, which can be used to verify if a string still maintains its membership with the language after being pumped. String generation is the retrieval of an ordered list of strings that belong to the language. This functionality is critical for validating that a significant number of strings in the language adhere to the conditions set by the pumping lemma. Lastly, as the name suggests, the minimum

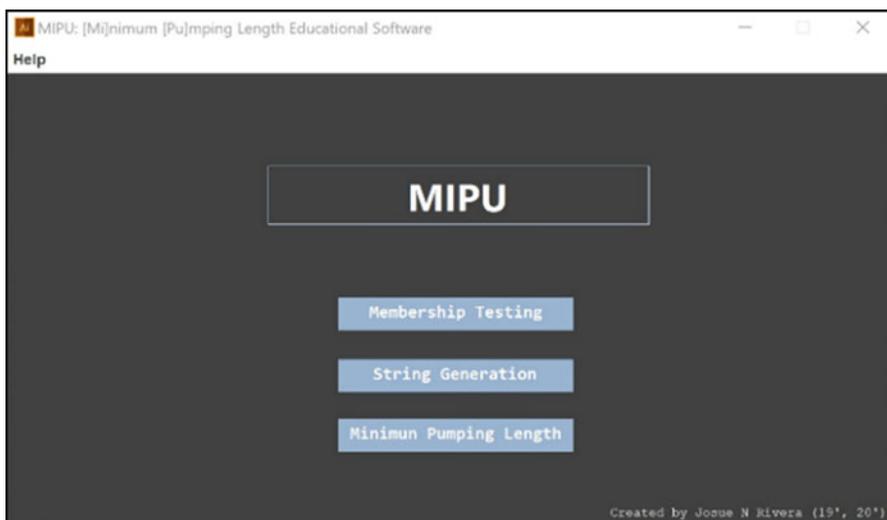


Fig. 3 Main menu of MIPU

pumping length determiner automatically calculates the minimum pumping length of a regular language described by a regular expression. It also, along with the minimum pumping length, provides the short strings that meet the conditions of the pumping lemma and the ways how the strings can be partitioned into three appropriate substrings x , y and z . These are core concepts that encompass the tools needed to determine the non-regularity of certain language using pumping lemma.

4.1 Membership Testing for Regular Languages

The membership testing module is composed of the regular expression to NFA converter and the NFA class described in Sects. 3.2.1 and 3.2.2, respectively. The core of the functionality is found in the “compute” method of the NFA class. The method traverses a graph created during the conversion of the regular expression to an NFA and observes if there is a path leading to an accept state.

As shown in Fig. 4, MIPU allows one to enter a regular expression and an input string. Then it takes the regular expression and generates an NFA for it. While computing membership, the input string is passed as a parameter to the NFA’s “compute” function, which returns either “True” or “False,” indicating whether the string belongs to the language or not.

Figure 5 presents another example for membership testing, where the regular expression is $(1\cup 0)^*101(1\cup 0)^*$ and the input string is 1011. As the result shows, the input string is determined to be a member of the language. The substring 101 of the given string reflects the segment 101 of the regular expression, while the symbol

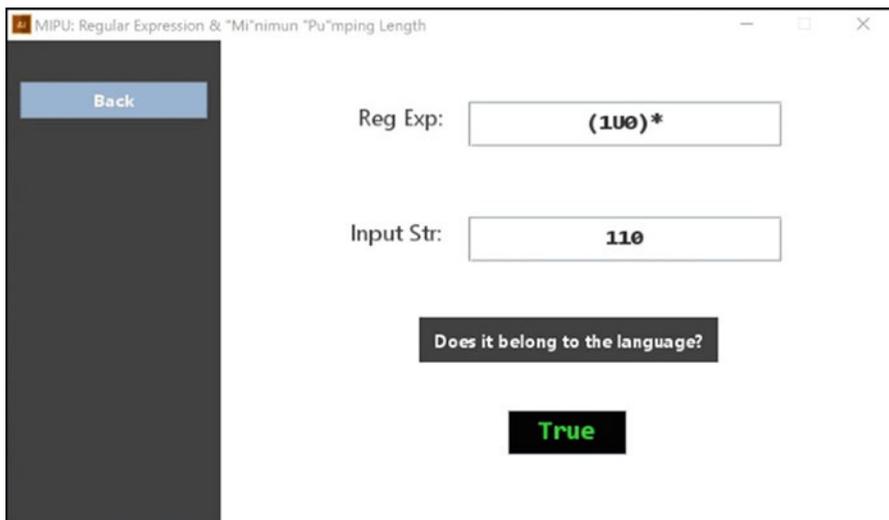


Fig. 4 Membership testing window after a string is tested

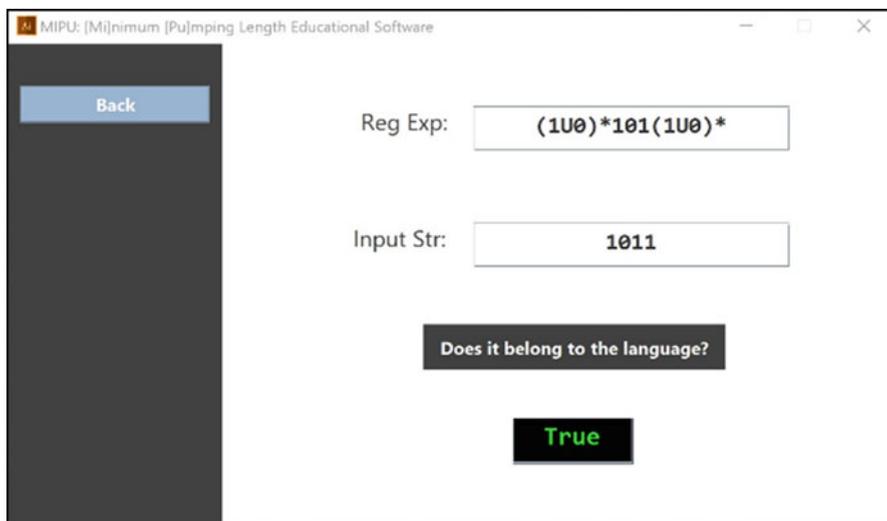


Fig. 5 Another example for membership testing

“1” at the end of the input string is the one generated by the rightmost segment $(1 \cup 0)^*$. Due to the tool’s ability to track multiple paths of the NFA as it computes a string, the only path that leads to an accept state for 1011 can be identified to accept the string.

4.2 String Generation

String generation for a given a regular expression is the second tool offered by MIPU. It is responsible for producing strings that are members of the regular language. The resulting strings are ordered by the length of the strings from the shortest to the longest. The generator can dynamically generate more strings as requested. This functionality uses the following components: regular expression to NFA converter, the NFA class, and the LSG. The LSG module uses the NFA produced from the regular expression and generates the strings from permutations of its alphabet that are members of the language. Various optimizations are used to eliminate branches of a permutation that will not lead to a valid string.

The string generation tool allows a user to enter a regular expression in the provided text field. After the regular expression is converted into an NFA, an LSG instance is created to generate strings that are recognized by the NFA. The LSG module dynamically calls a “generate” function that produces new strings as requested. Figure 6 shows some resulting strings after the “Get Strings” button is pressed. The generated strings belonging to the language are listed in a lexicographic

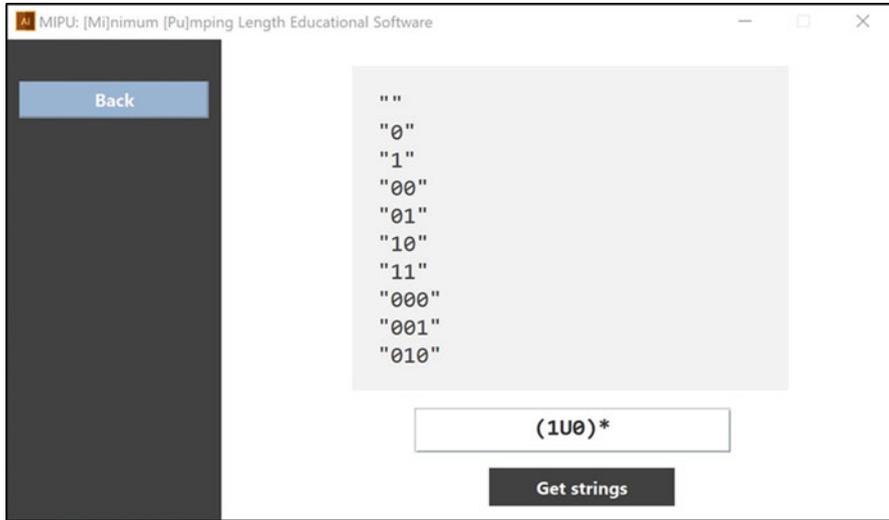


Fig. 6 An example of generating short strings

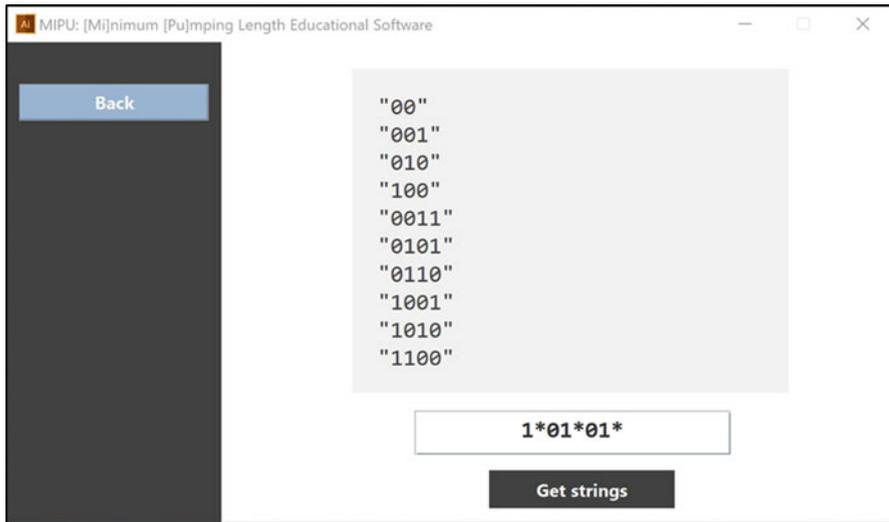


Fig. 7 Another example for string generation

order, which is the same as the dictionary ordering except that shorter strings precede longer ones.

Figure 7 shows another example for string generation. Note that the shortest string "00" is generated first by ignoring the segments containing a star operation. Then the following strings are generated by considering the segments containing a star operation, e.g., the last "1*" segment.

4.3 Determination of Minimum Pumping Length

The last function implemented in MIPU is to automatically calculate the minimum pumping length of a regular language. All modules of the MIPU framework, including conversion of a regular expression into an NFA and testing the various pumping lemma conditions, are used to achieve this function. As shown in Fig. 8, the minimum pumping length determination tool requires only a regular expression as its input. Once a regular expression is put in, an instance of the minimum pumping lemma determiner is created, which tests a significant number of strings belonging to the language and then decides the minimum pumping length. The figure shows that when the regular expression “ 10^*1 ” is typed in and the “Get Min Pump” button is pressed, the tool displays the minimum pumping length of the regular language along with a string example “101” that helps explain a way of portioning of the string that satisfies the pumping lemma conditions.

Figure 9 shows the minimum pumping length of the regular language 1^*01^*01 . In this scenario, the minimum pumping length is 3, and one of the minimum strings that meet the conditions of the pumping lemma property is 001. A possible partition of the string is also displayed. It should be noted that although 001 is selected, other minimum strings also exist, e.g., 100 and 010. One aspect of the results produced that should also be highlighted is the minimum string 001 given in Fig. 9 in comparison to the shortest string 00 shown in Fig. 7. In both scenarios, the regular expressions are the same, but the shortest string generated in Fig. 7 cannot be pumped; thus, it is not listed as a minimum string.

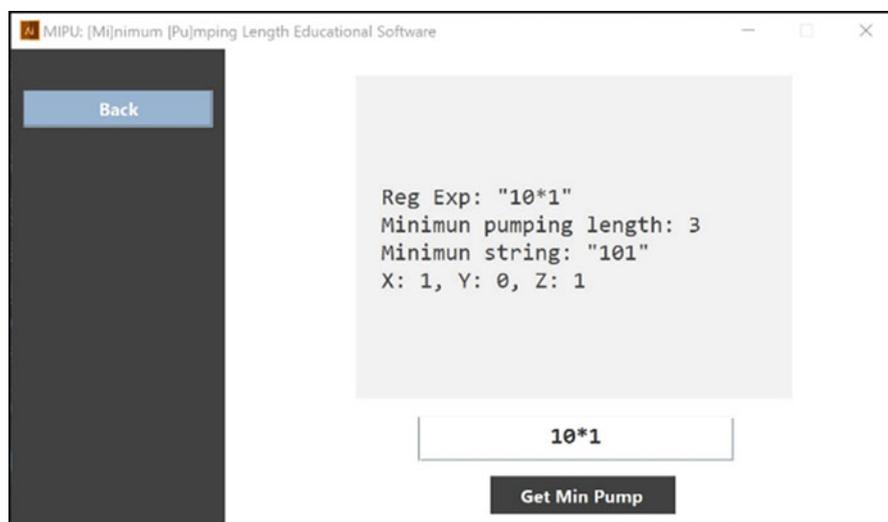


Fig. 8 Minimum pumping length determination

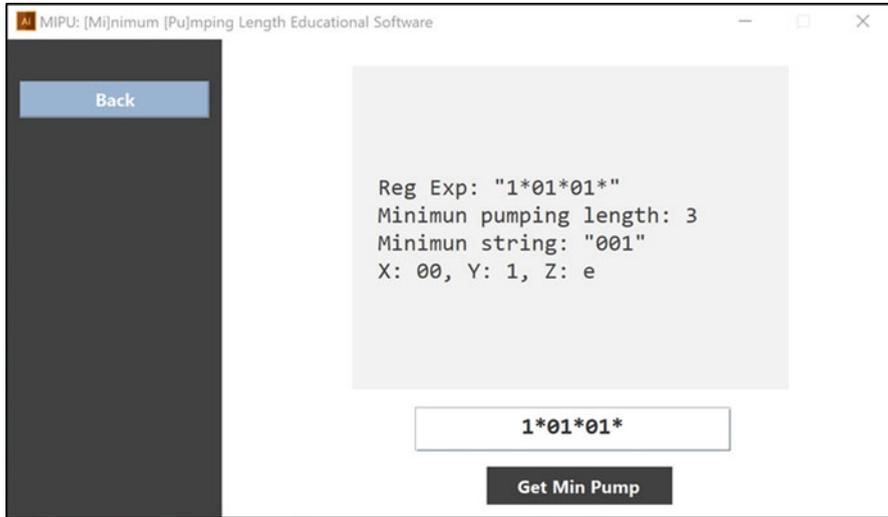


Fig. 9 Another example of minimum pumping length

One last example of minimum pumping length, illustrated in Fig. 10, is the regular expression $aabUa^*b^*$. The result is interesting because normally with a union operation where the left segment of the union operation represents a finite language and the right segment represents an infinite language, the minimum pumping length would be larger than the length of the finite segment since the string represented by the finite segment usually cannot be pumped. However, in this particular example, because the left segment can be generated by the right segment, the minimum pumping length of the regular expression equals to the minimum pumping length of the right segment, which is 1.

For more examples, the MIPU as well as the source code can be downloaded from the GitHub repository at <https://github.com/JosueCom/MIPU>.

5 Conclusions and Future Work

Finite automata and regular languages have brought humanity to a new age of innovation. They have led to advancements in artificial intelligence, the design of modern computers, and the representation of complex systems by a machine with limited memory. Through the MIPU project as well as the forthcoming improvements to enhance active learning, students will become more familiar with the formal concept of pumping lemma and overcome the complex challenge of understanding the concepts of regularity and non-regularity of languages. MIPU creates an environment that enables students to be actors for developing higher-

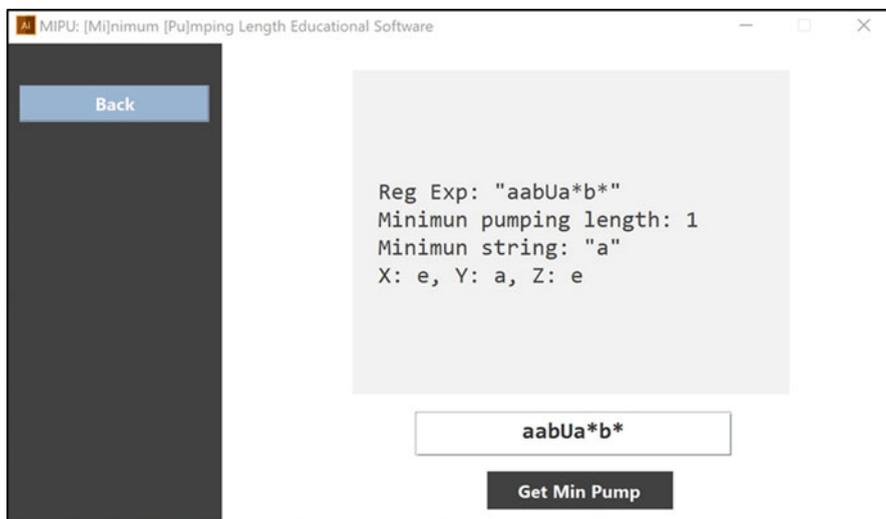


Fig. 10 One more example of minimum pumping length

order thinking and has the potential to be an effective tool in aiding students to better understand complex concepts.

For future work, we will improve MIPU to support visualization of the process of creating an NFA from a regular expression. We will also provide a pumping operation function that can retrieve a string that has been pumped for a given number of times. Additionally, the tool will allow a user to configure settings including redefining the restricted characters used to represent special symbols in a regular expression. The performance of generating strings may also be improved by designing a new generator that traverses the NFA graph when forming new strings instead of creating a permutation tree. Finally, we will redesign the GUI for string generation to allow dynamic generation of new strings when requested by users.

References

1. S. Yu, Regular languages, in *Handbook of Formal Languages*, Word, Language, Grammar, ed. by G. Rozenberg, A. Salomaa, vol. 1, (Springer, 1997), p. 41
2. W.S. McCulloch, W. Pitts, A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys.* **5**, 115–133 (1943)
3. G. Piccinini, The first computational theory of mind and brain: A close look at McCulloch and Pitts's 'logical calculus of ideas immanent in nervous activity'. *Synthese* **141**(2), 175–215 (2004)
4. M. Mani, N. Alkabout, D. Alao, Evaluating Effectiveness of Active Learning in Computer Science Using Metacognition, in *Proceedings of the 2014 IEEE Frontiers in Education Annual Conference (FIE'14)*, (Madrid, 2014), pp. 1–8
5. W.B. Wood, Clickers: A teaching gimmick that works. *Dev. Cell* **7**(6), 796–798 (2004)

6. K. Kim, P. Sharma, S. Land, M. Furlong, Effects of active learning on enhancing student critical thinking in an undergraduate general science course. *Innov. High. Educ.* **38**(3), 223–235 (2013)
7. H. Wang, Integrating modern software tools into online database course, in *Proceedings of the International Conference on Frontiers in Education: Computer Science and Computer Engineering (FECS'17)*, (Las Vegas, Nevada, 2017), pp. 100–103
8. S.H. Rodger, T.W. Finley, *JFLAP – An Interactive Formal Languages and Automata Package* (Jones and Bartlett Publishers, 2006)
9. M. LoSacco, S.H. Rodger, FLAP: A tool for drawing and simulating automata, in *ED-MEDIA 93, World Conference on Educational Multimedia and Hypermedia*, (1993), pp. 310–317
10. M. Procopiuc, O. Procopiuc, S. Rodger, Visualization and interaction in the computer science formal languages course with JFLAP, In *Proceedings of the 1996 Frontiers in Education Annual Conference (FIE'96)*, Salt Lake City, Utah, 6–9 Nov. 1996, pp. 121–125
11. M. Sipser, *Introduction to the Theory of Computation* (3rd Edition, Cengage Learning, 2013)

An Educational Guide to Creating Your Own Cryptocurrency



Paul Medeiros and Leonidas Deligiannidis

1 Introduction

Over the course of the past decade, many online transactions often required what is known as an “intermediary”—a third party that guarantees the secure exchange of both the goods and information pertaining to the transaction. Additionally, this means that all accountability for the transaction would fall into the hands of the third-party intermediary. This type of security framework is known as a “centralized” framework, as a central authority is responsible for executing a safe data exchange within user transactions. Contrarily, a “decentralized” security framework focuses on eliminating the intermediary, and instead using a “public ledger”—a database of all transaction records shared with all users. This method of transaction allows for the exchanged data between users to become immutable and cryptographically sealed. Additionally, the use of “ledgers” eliminates the chances of losing crucial information during a transaction, giving its users not only immense privacy and security capabilities but also great transparency with all of the transaction data. This type of decentralized security framework is what is used as the foundation for blockchain technology as it is known today. The most common form of blockchain technology, known as “cryptocurrency,” utilizes this framework by making all transaction history available to all users while also making all of its data immutable. Each time a series of new transactions is made within the blockchain, a new block containing the new transaction data will be created and added to the existing blockchain, further adding to the long list of immutable data. For this data to be accepted by the blockchain, it must be validated by the blockchain users

P. Medeiros · L. Deligiannidis (✉)

Department of Computer Science and Networking, Wentworth Institute of Technology, Boston, MA, USA

e-mail: medeirosp@wit.edu; deligiannidisl@wit.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_12

163

themselves through the use of a “proof-of-work” (PoW) algorithm. Miners run the PoW algorithm. This time-consuming process involves the solution to a hard problem [1] involving the computation of hashes which are one-way mathematical functions. While it is very hard to produce the correct hash for a block to be accepted into the blockchain, it is very simple and easy to verify the validity of the hash by any member of the blockchain community.

This report is primarily focused on the creation and deployment of a unique cryptocurrency while utilizing an existing codebase. Unlike the early stages of cryptocurrency that was still in its infancy, the information regarding its development process, especially its use of blockchain technology, has been greatly elaborated upon to make the deployment of a new cryptocurrency more streamlined. Producing a strong blockchain requires several different constituencies such as software developers, miners, exchanges, merchant processing services, web wallet companies, and user/consumers [2]. The process demonstrated here provides a streamlined approach to achieve all the necessary steps in deploying a fresh blockchain without the need of a large software developing team, or even large amounts of third-party software. The primary tools that are utilized in this development process largely center around the use of Bash, a Linux-based scripting language, and the Ubuntu operating system. Additionally, a GitHub account will be especially useful for pushing and pulling new builds of the cryptocurrency when needed.

2 A Working Codebase

We utilized a working cryptocurrency codebase to create our own new cryptocurrency. This codebase allows for anyone to utilize the binaries provided by it to compile their own unique cryptocurrency build. More specifically, this codebase gives developers the necessary tools needed to deploy their own cryptocurrency. Of course, this codebase is not able to generate a new cryptocurrency by itself—there are many necessary alterations of code and unique configuration parameters that must be provided for a new cryptocurrency to be built properly using this codebase. For this project, we used the Litecoin codebase, which is an open-source peer-to-peer cryptocurrency project, and we utilized the public Litecoin GitHub repository for its source code. To gain access to the “Litecoin” repository and fork the necessary documents, a GitHub account needs to be created and then linked to the Ubuntu operating system (which can be done via the Bash terminal setup in Visual Studio Code). This connection with GitHub and Ubuntu is necessary, because if data needs to be pulled and/or be updated from the new forked repository, it can be done through the Ubuntu command line via bash scripting, without having to access the GitHub website directly.

To get the working codebase, we first installed the Ubuntu subsystem on a Windows 10 machine and forked the Litecoin repository from <https://github.com/litecoin-project/litecoin> and then followed the instructions of the “build-unix” file located in the document folder to install the dependencies.

3 Preliminary Code Modifications

Renaming assets in an existing codebase as a step toward creating a “unique cryptocurrency” is not as questionable as it may seem. Many cryptocurrencies, much like Litecoin itself, often merge their code with changes made from other codebases. In many cases, Litecoin can be seen merging code from Bitcoin, a major cryptocurrency. This can be viewed when going to the Litecoin repository page and checking its commit history and changes. Unsurprisingly, both Bitcoin and Litecoin share enough similarities in their source code to make merging code possible and not very tedious. Practicing something similar to Litecoin, portions of the codebase installed via the Bash terminal can be renamed to include the developer’s new cryptocurrency name. Moving forward, it is important to note that there are other aspects of the codebase that must be edited that stretch beyond simply renaming things. The commands shown in Fig. 1 should be entered into the Bash terminal to replace all areas of the code where Litecoin is mentioned.

An imaginary cryptocurrency named “CloudCoin” is used in this example to demonstrate what should be edited in the respected fields. In Fig. 1, any instance of “CloudCoin” should be replaced with the unique cryptocurrency name of the developer’s liking. Additionally, the abbreviation of “CloudCoin,” which is “CLC,” should also be changed to an abbreviation of the new cryptocurrency’s name.

Additionally, there are two more name changes that must be made. In Litecoin, and other cryptocurrencies, there are monetary denominations used to represent amounts of cryptocurrency that are smaller than a single coin. In Litecoin, these two denominations are known as “lites” and “photons.” Figure 2 shows how to replace these denominations with our own called “clouds” and “raindrops,” respectively.

Next, locate the file “chainparams.cpp.” This file is one of the major pieces in creating a new blockchain and requires a variety of edits to make a successful

```
find ./ -type f -readable -writable -exec sed -i "s/Litecoin/Cloudcoin/g" {} \;
find ./ -type f -readable -writable -exec sed -i "s/LiteCoin/CloudCoin/g" {} \;
find ./ -type f -readable -writable -exec sed -i "s/LTC/CLC/g" {} \;
find ./ -type f -readable -writable -exec sed -i "s/litecoin/cloudcoin/g" {} \;
find ./ -type f -readable -writable -exec sed -i "s/litecoind/cloudcoind/g" {} \;
```

Fig. 1 The commands to replace any instances of the word “Litecoin” or its variations within the source code. It is important to replace the examples of “CloudCoin” with the developer’s own unique cryptocurrency names

```
find ./ -type f -readable -writable -exec sed -i "s/lites/clouds/g" {} \;
find ./ -type f -readable -writable -exec sed -i "s/photons/raindrops/g" {} \;
```

Fig. 2 The commands run via the Bash terminal to replace any instances of the words “lites” or “photons”—the denominations used for Litecoin when the amount of currency is less than one ‘Litecoin’

```
pchMessageStart[0] = 0xfb;
pchMessageStart[1] = 0xc0;
pchMessageStart[2] = 0xb6;
pchMessageStart[3] = 0xdb;
```

Fig. 3 Four lines of code that represent the PCH Message Values that are present within the “chainparams.cpp” file. The bytes associated with these lines of code must be changed to unique values to ensure that the different networking protocols present in the blockchain can be successfully handled

```
base58Prefixes[PUBKEY_ADDRESS] = std::vector<unsigned char>(1,48);
base58Prefixes[SECRET_KEY] = std::vector<unsigned char>(1,176);
base58Prefixes[EXT_PUBLIC_KEY] = {0x04, 0x88, 0xB2, 0x1E};
base58Prefixes[EXT_SECRET_KEY] = {0x04, 0x88, 0xAD, 0xE4};
```

Fig. 4 Four lines of code that represent the different public and secret keys that will be present in the cryptocurrency blockchain. All of these values must be unique so that the blockchain can receive accurate data from its users

build. Search in the file for several lines of code that start with the phrase “pchMessageStart,” followed by a series of bytes. The bytes present and handle different networking protocols being used to identify the clients of the blockchain. These values must be changed to something unique, because if another cryptocurrency uses the same PCH message values, it will create complications when attempting to identify which cryptocurrency blockchain it is trying to access. The section of code that should be edited should look like Fig. 3. Note that the bytes supplied in the figure are the default values given by Litecoin and should not be used as input.

Like the previous step, the next section of code is also present within the “chainparams.cpp” file but begins with the phrase “base58Prefixes.” The bytes associated with these lines of code are used as prefixes for the addresses that can receive data from the cryptocurrency blockchain users. These values must be unique, since sharing these addresses with another cryptocurrency can confuse which cryptocurrency blockchain the data will be sent to. The four lines in Fig. 4 hold the address data for the public key and secret key (as well as the external public and secret keys). The values present within the figure should not be used as input for the code and instead should serve as an example as to the possible values that could be used.

4 Creating the Genesis Block

This is arguably the most important section of the development process, as the following steps are used for creating the first block of the new cryptocurrency blockchain. The first block of a new blockchain, otherwise known as the “genesis

block,” is essentially the “origin” of a new cryptocurrency’s blockchain. It plays a crucial role not only in creating the new blockchain itself but also for allowing successive blocks to be created and added in the chain. The data structure that exists within each block of the chain is known as the “Merkle root” and must be created alongside the genesis block. The Merkle root consists of what are called “chained hashes.” Inside of the “chainparams.cpp” file, the developer can find examples of the genesis block and Merkle root values.

Thankfully, a Python script exists that can help assemble these necessary pieces of data to generate a successful genesis block for the developer. The script, known as “GenesisH0,” can be found on GitHub and was utilized for the sake of creating a unique genesis block. Figure 5 shows how to download and install GenesisH0.

In the install directory, there is a python script named “genesis.py,” which is the script that will be calculating the nonce and assembling the additional information to create the genesis block for the new blockchain. To use the script, enter the following command shown in Fig. 6 into the terminal—substituting the placeholders with the unique values for the article, public key, and timestamp obtained above (as well as an arbitrary number for the nonce). Be sure to include quotations around certain values as shown in Fig. 6.

It is important to note that this (mining) process can take a long time to complete, as searching for a suitable nonce can be very difficult; we found one within 48 hours. Sometimes, this process can take minutes, while other times, it can take several hours to complete. If the developer finds that they cannot successfully obtain a suitable nonce, either change the arbitrary value given to the nonce or allow the script to run for a longer time.

When the developer sees the message indicating that the genesis hash is found, copy down the nonce and genesis hash values that appear in their respective results. Re-run the previous python script command in Fig. 6—using the new nonce value as the nonce parameter for the script. Running this command should immediately return a result that looks similar to Fig. 7. Values associated with data such as the “Merkle hash,” “bits,” and other outputs will have unique values when run through the developer’s terminal. It is crucial to take note of the values associated with all outputs of the script. Pay close attention to the “Merkle hash,” “pubkey,” “time,”

```
git clone https://github.com/lhartikk/GenesisH0.git
sudo pip install scrypt construct==2.5.2
```

Fig. 5 Two commands run via the Bash terminal. The first installs the necessary Python script associated with “GenesisH0,” while the second installs dependencies that allow the Python script to run

```
python genesis.py -a scrypt -z "Insert Article Here" -p "Public Key" -t timestamp -n nonce
```

Fig. 6 A sample input command for the “genesis.py” script to find a suitable nonce. All placeholders should be substituted for their real corresponding values

```
python genesis.py -a scrypt -z "Insert Article Here" -t timestamp -n nonce
algorithm: scrypt
merkle hash: merkle-hash-value
pszTimestamp: Article-Website Date Title
pubkey: public-key
time: unix-time-value
bits: bit-value
Searching for genesis hash..
genesis hash found!
nonce: nonce-value
genesis hash: genesis-hash-value
```

Fig. 7 The output of the “genesis.py” script when the new suitable nonce is used as the nonce parameter for the command

“bits,” “nonce,” and “genesis hash” values. This information will aid in developing the next step of the code development process.

5 Primary Code Modifications

Now that the data required to create a new genesis block has been obtained, we need to navigate back to the “chainparams.cpp.” First, edit the value associated with the variable “pszTimestamp,” and replace it with the name of the article the developer used to create the previous genesis block data.

Scrolling further down the file, there should be a class called “CMainParams.” Within it, there is a line of code that refers to creating a genesis block, as well as two lines of code that begin with the word “assert,” followed by the words “hashGenesisBlock” or “hashMerkleRoot.” The first line of code should include the phrase “CreateGenesisBlock” that holds three genesis block values associated with its Unix time, nonce, and bits. Modify these values with the new values generated from the execution of the “genesisH0” script.

Below the “CreateGenesisBlock” line, the value associated with the line “assert(consensus.hashGenesisBlock)” should be modified to include the genesis hash value the developer obtained from the “genesisH0” script. Additionally, the value associated with “assert(genesis.hashMerkleRoot)” should be modified to include the new Merkle hash.

Cryptocurrencies are typically known to have what are called “decentralized security frameworks.” This type of framework eliminates the need for “intermediaries,” which are responsible for guaranteeing a secure exchange of data (in this case, money) between the users executing a transaction [3]. A “public ledger” is put

in place of the “intermediary,” which is an immutable, cryptographically secured permanent record of all transactions among all users of the blockchain [3].

While this provides a secure alternative to intermediaries, its main strengths lie in its ability to eliminate the chances of information loss, having powerful transaction validation abilities, easy verification processes, and a strong focus on transaction transparency [3]. Interestingly, Litecoin (as well as similar cryptocurrencies, such as Bitcoin) references “dnsseeds” and “seednodes” in its source code, which means that there are multiple active IP addresses that are running to support client interactions with Litecoin (such as transactions). In a way, these could be seen as a form of “intermediaries,” but they are in no way required to set up a fresh cryptocurrency blockchain. It’s crucial to remove these unnecessary pieces of data, as including them in the new blockchain will send clients of the new blockchain to the addresses provided by the Litecoin source code.

To begin removing the “dnsseeds,” navigate back to the “chainparams.cpp” file. Toward the bottom of the file, several lines of code that begin with the phrase “vSeeds.emplace_back” should be present. The developer can either choose to comment out these lines or delete them (doing either will disable these lines of code). Within the same “src,” open the “chainparamsseeds.h” file. Edit the method referred to as “pnSeed6_main” by commenting (or deleting) all its accompanying data. The data present in this method is memory associated with the nodes used by the Litecoin source code. Specifically, each line of this method contains data for a unique IP address associated with Litecoin, alongside a port number used by the accompanying address. Because nodes for the new blockchain have not been set up yet (nor can they use the same values provided by Litecoin), these values must be removed from the source code. Additionally, the developer should make sure that the method below “pnSeed6_main,” named “pnSeed6_test,” is left alone.

6 Deploying the Nodes

Here, a peer-to-peer (P2P) network is used to establish the ability for clients to mine, send, and receive cryptocurrency from the new blockchain. Using this P2P network, the transactions and blocks made through the blockchain will be broadcasted by the nodes and sent to their peers, which then relay further to flood the network if they meet the relay policies [4]. In other words, the P2P network serves as a component that protects its users from “Denial of Service” attacks (DoS) in addition to supporting transactions through Simple Payment Verification (SPV) [4]. The tools used to accomplish this goal were provided through the Microsoft Azure platform and its ability to rapidly deploy multiple virtual machines.

Like Bitcoin, the users and/or computers that will be running one (or multiple) of these nodes will have a direct and authoritative view of the blockchain, with a local copy of all the transactions, independently validated by their own system [5]. This means that if the developer chooses to use their own personal nodes instead of using a service such as Microsoft Azure, then the developer can view the entire history

of the blockchain with other additional privileges. However, running personal nodes will require a permanently connected system in which the system must have enough resources to process all the blockchain transactions [5]. It should also be noted that there may be situations in which two nodes may broadcast different versions of the next block of data simultaneously—which will cause some nodes to receive one or the other versions first [6]. This does not mean anything negative has occurred, but the nodes will continue to compute the work they have been given until the block with the largest amount of work (“largest branch,” “longest chain”) is identified—to which the other nodes will switch to the branch with the largest amount of work completed [6].

Once the developer has navigated to the Microsoft Azure portal, there are many options that Microsoft provides to its users to deploy a variety of different technologies. One such option is “virtual machines.” After selecting the “virtual machines” option, select the “add” option on the page to bring up the setup process for the first virtual machine (these virtual machines will be used as the nodes for the new blockchain). It is recommended when setting up any of the virtual machines to set the virtual machine operating system as Linux, as well as having it run version 16.04 of Ubuntu. After successfully deploying the first virtual machine, a second one with the exact same parameters should also be deployed.

Once both virtual machines are deployed, selecting any of the virtual machines should display information regarding its network protocol, as shown in Fig. 8. Each line should be present in the “inbound port rules” section of the network protocol of the virtual machine, except for the first line. The first line (the lined called “Port_9444”) must be manually added to both the “inbound port rules” and the “outbound port rules” of both virtual machines.

To do so, select the “add inbound port rule” option on the page, and change the “destination port ranges” value to the default port number associated with the new blockchain.

Additionally, the developer should change the priority value to 100, as well as the “Name” of the security rule to the default port number. Figure 9 provides an example of what the sample inputs should look like for both virtual machines. It

Priority	Name	Port	Protocol	Source	Destination	Action
100	Port_9444	9444	Any	Any	Any	Allow
65000	AllowVnetInBound	Any	Any	VirtualNetwork	VirtualNetwork	Allow
65001	AllowAzureLoadBalancerInBound	Any	Any	AzureLoadBalan...	Any	Allow
65500	DenyAllInBound	Any	Any	Any	Any	Deny

Fig. 8 An example demonstrating the correct input/output port rules associated with both of the deployed virtual machines. The first rule, titled “Port_9444,” should be missing after initially deploying both of the virtual machines and must be added manually

Fig. 9 A sample input for the additional inbound/outbound port rule that must be added to both virtual machines

The screenshot shows the configuration page for a port rule named "Port_9444" in the Azure portal. At the top, there are buttons for "Save", "Discard", "Basic", and "Delete". The configuration fields are as follows:

- Source:** A dropdown menu set to "Any".
- Source port ranges:** A text input field containing an asterisk (*).
- Destination:** A dropdown menu set to "Any".
- Destination port ranges:** A text input field containing "9444".
- Protocol:** A set of radio buttons with "Any" selected, and options for "TCP", "UDP", and "ICMP".
- Action:** A set of radio buttons with "Allow" selected, and an option for "Deny".
- Priority:** A text input field containing "100".
- Name:** A text input field containing "Port_9444".
- Description:** A text input field containing "Crypto".

is also important to note that the “outbound port rules” should be identical to the inbound port rules.

In order for clients of the blockchain to receive updates and submit transactions, they must know the proper nodes and ports to connect to the blockchain. This can easily be done by adding a “.conf” file to the root directory of the project (create a “.conf” file in the “litecoin” directory, which should be the directory that holds all of the files for the developer’s current build). The .conf file should be titled whatever the name of the developer’s cryptocurrency is. Change the values for the “addnode” section, and supply them with the correct information provided by the Microsoft Azure portal. Typically, the format for the “addnode” values is the name of the node (in this case, the virtual machine’s name), the node’s location, the phrase “.cloudapp.azure.com.”, and the default port number. Additionally, the values for “rpcuser” and “rpcpassword” must be changed if the developer wishes to mine their cryptocurrency on a local build of their blockchain. A problem we encountered is the fact that there is no “.conf” file given by the Litecoin source code, meaning there’s no file to edit, like the other examples. Thus, in Appendix A, we share our own “.conf” file.

7 Building the Wallet

Now that the nodes and “.conf” file have been successfully created, the source code can finally be recompiled and built with wallet functionality. Enabling wallet functionality will compile the source code with a functional user interface that will allow the users of the cryptocurrency to both mine and exchange currency between one another. When using the digital wallet that will be compiled by Litecoin’s source code, each user (wallet) will receive a set of “keys” that will allow users to interact with each other’s wallets. The user’s “private key” is to sign and protect the information of the user’s wallet [7, 8]. If a user has the private key to an address (wallet), then that user can use that key to access the currency associated with that address from any Internet-connected computer [2, 7]. Litecoin’s source code includes the tools necessary for wallet functionality through the use of “QT,” an application designed for developing user interfaces. Compiling the new source code with the QT application provided by Litecoin produces a new executable file that will run the new cryptocurrency wallet. If the various user-interface assets are not updated to reflect the new cryptocurrency, they may still refer to Litecoin on the user interface. However, all transactions that take place in the executable file will still use the new cryptocurrency, so changing the names of the assets is not necessary for deployment. If the new cryptocurrency is intended for public use however, it’s recommended that the assets are updated to reflect the names and abbreviation of the new currency.

To build the wallet, run the “autogen.sh” and the “configure” scripts. This creates an executable file named “Litecoin.qt.” This file should be run to access the user interface of your wallet as shown in Fig. 10. It is also possible that the executable filename may also be named after your cryptocurrency name—and it is also possible that it could be misplaced in one of the source code subfolders upon compilation. If you cannot find the “.qt” file, the code should be recompiled.

If the Litecoin QT application successfully connects to the nodes, it is possible to locally mine the new cryptocurrency on the developer’s computer. Generating currency can also allow the developer to test transactions between users once enough currency has been generated. To begin mining currency, a simple executable file needs to be created with the code shown in Fig. 11. The developer should change the instance of “Litecoin” in the code to reflect the name of the new currency.

While there are no necessary steps left to take for producing a privately distributed build for testing purposes, there are several additional steps online that the developer may wish to follow to make managing the blockchain easier, as well as prevent potential security breaches. It is recommended that the developer research these additional steps if they wish to make their cryptocurrency available to the public.

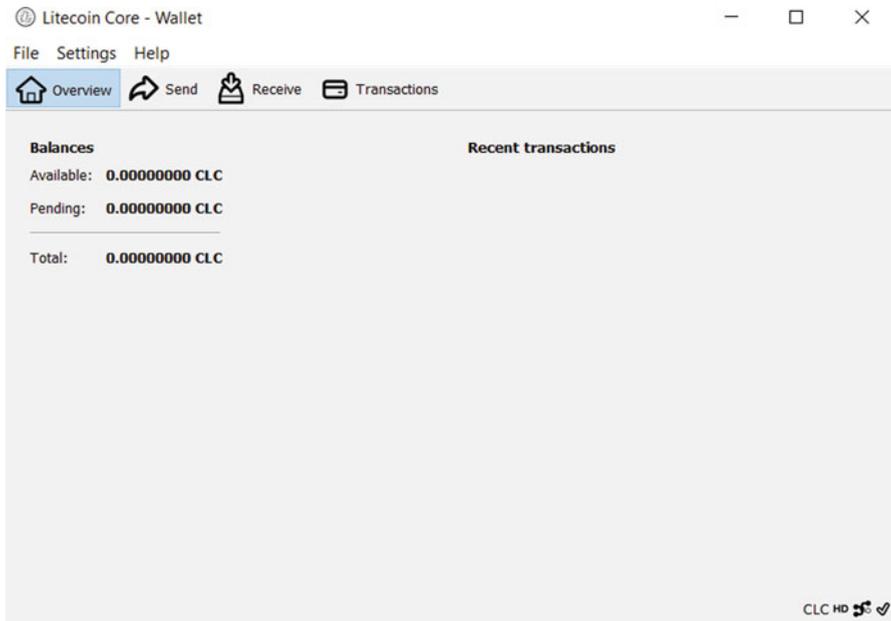


Fig. 10 Litecoin’s built-in QT wallet application. While all of the assets present in the user interface were not changed (such as the logo used for Litecoin and the header of the executable file saying “Litecoin Core”), the assets referring to the type of currency being used were updated to reflect the new currency, as referenced by the “CLC” abbreviation next to the balance. Normally, the abbreviation would be “LTC” to refer to Litecoin, whereas in this executable file, it was changed to “CLC” to reflect “CloudCoin,” an imaginary currency used for the sake of this study

```
#!/bin/bash
echo "Generating currency! (CTRL+Z to stop)"
while :
do
  litecoin-cli generate 1
done
```

Fig. 11 The code used to mine the newly compiled cryptocurrency

8 Results and Discussion

Scalability and security are two of the most important aspects of cryptocurrencies. With public interest of cryptocurrency rising, the possibility of encountering scalability issues has unfortunately become inevitable. In short, this problem refers to the capability of a single node on a blockchain network to handle a growing amount of transactions per second and thus be enlarged to accommodate that growth [9]. While there have been various attempts to combat this issue such as decreasing the block size or increasing the number of nodes operating with the blockchain, many of the potential solutions are expensive and potentially cost-ineffective.

Because of scalability issues, many cryptocurrencies are often limited to how many transactions they can handle at one time. There are several other factors that contribute to this limitation as well. One of the other factors that could be considered is the speed at which the transactions are completed and placed on the chain. Typically, this is determined by the amount of network activity taking place on the blockchain, alongside the transaction fees associated with the exchanging of the currency itself. When comparing against similar cryptocurrencies such as Bitcoin, Litecoin completes transactions around 4 times faster than Bitcoin. On average, Litecoin takes approximately 2.5 minutes to complete a single exchange, while Bitcoin takes approximately 10 minutes to complete the same task. Additionally, Litecoin can handle about 56 transactions per second as opposed to Bitcoin, which can only handle around 7 [10]. Litecoin is usually seen as a faster alternative to Bitcoin when it comes to exchanges on the blockchain, so this report decided to use the Litecoin codebase to provide an alternative solution for rapid development and experimenting that can support larger numbers of transactions at a time than Bitcoin.

Aside from transaction speed, the ability to provide a strong, secure method for users to interact with the blockchain can also be considered an extremely important aspect of development. In other words, the security of the blockchain is a major concern—and typically involves the confidentiality, integrity, and availability of the technology itself [9]. To satisfy these security concerns, both Bitcoin and Litecoin utilize what are called “proof-of-work” (PoW) algorithms to cryptographically seal the transactions in a block of the blockchain. In short, these algorithms prevent others from tampering with information in a block, providing a secure way of storing transactions in a block. While it is easy to verify the validity of the block or the entire blockchain, it is infeasible to modify a transaction without rerunning the PoW algorithm for each block in the chain!

While there are several different types of PoW algorithms that are used with various cryptocurrencies, the most immediate example would be Bitcoin, with its use of the SHA-256 hash algorithm. Litecoin’s source code holds many similarities with Bitcoin. However, one of its key differences includes Litecoin’s decision to use a PoW hash algorithm, *scrypt*, instead of SHA-256. Both algorithms aim to compute hashes of data present on the blockchain, as well as authenticate the transaction data that is stored in each block.

Both SHA-256 and *scrypt* hash functions are computationally inexpensive to run. However, there is no known way of generating a specific hash value based on some input. Miners try different combinations of nonce and rerun these hashing algorithms, and when a desired hash value is computed, they are awarded, and the block can be added in the blockchain. What makes it even harder is that *scrypt* is also memory intensive because the generated hashes are stored in memory, and then they need to be accessed before submitting a solution. This makes *scrypt* appealing since miners cannot use Application-Specific Integrated Circuits (ASICs) to mine hashes fast. *Scrypt* provides users with less-devoted hardware to be able to mine currency from the blockchain, as opposed to SHA-256 which requires users to join “mining pools” to cooperate in mining currency. This does not mean SHA-256’s methods

are completely safe, however. As stated by Chang, blockchain mining pools are also vulnerable to attacks in which the miner in a compromised pool withholds and delays blocks while submitting shares, effectively taking all of the rewards from the mining pool [11]. A precise definition of this occurrence would be what is called a “block withholding attack.” According to Kamhoua, these attacks are defined as the situations in which a miner decreases the expected revenue of a mining pool by withholding authenticated blocks—but also increases their own reward by submitting as many shares as possible to the pool [12]. The choice to use Litecoin for this study allows for a better testing environment upon initial deployment—but like working with any codebase, it will require improvements to security protocol and maintenance of several blockchain components if there are any attempts in making the cryptocurrency commercially viable.

Appendix A

An example “.conf” file created for clients of the blockchain to know how and where to receive updates and submit transactions. The file provides details concerning the proper nodes and ports to connect to the blockchain. The “.conf” file should be added to the root directory of the project.

```
#cloudcoin.conf configuration file.
# Network-related settings:
# Run on the test network instead of the real cloudcoin network.
#testnet=0
# Connect via a socks4 proxy
#proxy=127.0.0.1:9050
# Use addnode= settings to connect to specific peers
addnode=NODE1.eastus.cloudapp.azure.com:9444
addnode=NODE2.eastus.cloudapp.azure.com:9444
# Use connect= settings as you like to connect ONLY to
  specific peers:
#connect=localhost:9444
# Do not use Internet Relay Chat (irc.lfnnet.org #cloudcoin
  channel) to find other peers.
#noirc=0
# Maximum number of inbound+outbound connections.
#maxconnections=
# JSON-RPC options (for controlling a running cloudcoin/
  cloudcoind process)
# server=1 tells cloudcoin-QT to accept JSON-RPC commands.
server=1
# You must set rpcuser and rpcpassword to secure the JSON-RPC
  api
rpcuser=username123
rpcpassword=password123
# How many seconds cloudcoin will wait for a complete RPC HTTP
  request after the
# HTTP connection is established.
```

```

rpctimeout=30
# By default, only RPC connections from localhost are allowed.
  Specify as many rpcallowip= settings
# as you like to allow connections from other hosts (and you may
  use * as a wildcard character):
#examples:   rpcallowip=10.1.1.34   rpcallowip=192.168.*.*
  rpcallowip=1.2.3.4/255.255.255.0
rpcallowip=127.0.0.1
# Listen for RPC connections on this TCP port:
#rpcport=9432
# You can use cloudcoin or cloudcoind to send commands to
  cloudcoin/cloudcoind
# running on another host using this option:
#rpcconnect=192.168.2.29
# Use Secure Sockets Layer (also known as TLS or HTTPS)
  to communicate with
# cloudcoin -server or cloudcoind
#rpcssl=1
# OpenSSL settings used when rpcssl=1
#rpcsslciphers=TLSv1+HIGH:!SSLv2:!aNULL:!eNULL:!AH:!3DES:
  @STRENGTH
#rpcsslcertificatechainfile=server.cert
#rpcsslprivatekeyfile=server.pem
# Miscellaneous options. Set gen=1 to attempt to generate
  cloudcoins
gen=1
# Use SSE instructions to try to generate cloudcoins faster.
4way=1
# Pre-generate this many public/private key pairs,
  so wallet backups will be valid for both prior
# transactions and several dozen future transactions.
#keypool=100
# Pay an optional transaction fee every time you send
  cloudcoins. Transactions with fees are more likely
# than free transactions to be included in
  generated blocks, so may be validated sooner.
paytxfee=0.001
# Allow direct connections for the 'pay via IP address' feature.
#allowreceivebyip=1
# User interface options
# Start cloudcoin minimized
#min=1
# Minimize to the system tray
#minimizetotray=1
#THIS IS THE END OF THE FILE.

```

References

1. Bitcoin Wiki: Difficulty <https://en.bitcoin.it/wiki/Difficulty>. Retrieved 2 Feb 2020
2. M. Swan, *Blockchain – Blueprint for a New Economy* (O'Reilly Media Inc., 2015) ISBN-13: 978-1491920497

3. D. Puthal, N. Malik, S.P. Mohanty, E. Kougianos, C. Yang, The Blockchain as a Decentralized Security Framework. IEEE Consumer Electronics Magazine, 18–21 (2018)
4. I. Giechaskiel, C. Cremers, K.B. Rasmussen, When the Crypto in Cryptocurrencies Breaks: Bitcoin Security under Broken Primitives. IEEE Computer and Reliability Societies (2018)
5. A.M. Antonopoulos, *Mastering Bitcoin, 2nd Edition*. ISBN: 9781491954386 (O'Reilly Media Inc., 2017)
6. S. Nakamoto, Bitcoin: "A Peer-to-Peer Electronic Cash System". <https://bitcoin.org/bitcoin.pdf>. Retrieved 3 Feb 2020
7. K.A. Taher, T. Nahar, S.A. Hossain, Enhanced Cryptocurrency Security by Time-Based Token Multi-Factor Authentication Algorithm. International Conference on Robotics, Electrical and Signal Processing Techniques (ICREST) (2019)
8. J. Song, *Programming Bitcoin. Learn How to Program Bitcoin from Scratch* (O'Reilly Media Inc., 2019) ISBN-13: 978-1492031499, 2017
9. G. Sargsyan, N. Castellon, R. Binnendijk, P. Cozijnsen, Blockchain Security by Design Framework for Trust and Adoption in IoT Environment. IEEE World Congress on Services (SERVICES) (2019)
10. Which Cryptocurrencies Have The Fastest Transaction Speeds? International Business Times [U.S. ed.], 2018. Gale Academic OneFile, https://link.gale.com/apps/doc/A523776350/AONE?u=wit_main&sid=AONE&xid=ea2a13f9
11. S.-Y. Chang, Y. Park, Silent Timestamping for Blockchain Mining Pool Security. 2019 Workshop on Computing, Networking and Communications (CNC)
12. D.K. Tosh, S. Shetty, X. Liang, C.A. Kamhoua, K.A. Kwiat, L. Njilla, Security Implications of Blockchain Cloud with Analysis of Block Withholding Attack. 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (2017)

Peer Assistant Role Models in a Graduate Computer Science Course



Evava Pietri, Leslie Ashburn-Nardo, and Snehasis Mukhopadhyay

1 Introduction

Because information technology has transformed nearly every field of human endeavor, it is difficult to imagine a career path that will not involve some interaction with computers and computing. Consequently, the predicted job growth in computer-focused careers from the years 2014 to 2024 is very high (12.5 percent)—higher than all STEM fields combined [3]. High schools and colleges in the USA are struggling to keep pace with this need, leading the National Science Foundation and the US Government to outline a plan to provide large-scale support to states, schools, and universities to further CS education [44]. Between 2008 and 2014, there was a 53% increase in the population of international students in science and engineering departments in the USA [27]. To meet the CS demand, it is therefore imperative to recruit from groups that have been traditionally underrepresented in these fields (e.g., women; [28]) as well as from international students [35]. Beyond helping to fulfill workforce needs, diverse research teams in STEM also produce to higher quantity and quality of outputs [9, 30].

One strategy to help recruit underrepresented students into CS majors and the CS workforce is to ensure students take classes with professors from these groups [47]. That is, these professors can act as role models (i.e., a person to which one feels similar and aspires to be like; [12]) to inspire students' interest in CS careers [1]. Unfortunately, CS remains one of the least diverse areas in academia (particularly

E. Pietri (✉) · L. Ashburn-Nardo
Department of Psychology, IUPUI, Indianapolis, IN, USA
e-mail: epietri@iupui.edu; lashburn@iupui.edu

S. Mukhopadhyay
Department of Computer and Information Science, IUPUI, Indianapolis, IN, USA
e-mail: smukhopa@iupui.edu

with regard to gender; [28]), and thus, students from underrepresented groups may have few opportunities to interact with professors who would be most inspirational. Moreover, ensuring students learn from professors matching their identities (e.g., female students can interact with female professors) may create heavy service expectations and harm the research productivity of the few female computer scientists in academia [48]. Thus, it is critical to test new techniques that will help recruit underrepresented groups into academic CS and diversify professors in CS. The current work aims to address this national need, by testing a new intervention to recruit international students and, particularly, international female students into CS PhD programs. We specifically implemented our intervention in a Master's level course, which has a high proportion of female and international students and, thus, represents an opportunity to recruit students from underrepresented groups into academic CS. Moreover, the Master's level represents a critical junction for students' future career trajectory. That is, Master's students can decide to finish with a terminal Master's degree and pursue an industry profession, or these students can continue on to a PhD program in CS and seek an academic job. Thus, this Master's course is a prime opportunity to test a novel intervention.

There are a variety of reasons why international female Master's students may choose not to pursue an academic career in CS. Most germane to the present research are their perceived fit in CS and their perceived future selves. Specifically, relative to US-born female students, international female students may have less exposure to the masculine stereotypes about CS that are pervasive in the USA [4, 29], but they still may have self-concepts (i.e., knowledge and beliefs about themselves) doubting their fit in academic computer science, and their possible future selves (i.e., their representations of who they could become in the future) may not include academic computer scientist researchers [22, 23]. People use culture, important groups, and families as sources of information about their self-concept and possible future selves [20, 39, 41]. Because of cultural expectations and familial obligations, international female Master's students may perceive themselves as computer programmers destined to work in the industry in order to provide for their families rather than as academic researchers with more individualistic achievement goals [20]. These self-concepts and possible future selves in turn may lead to four critical downstream consequences impeding international female students' pursuit of academic CS. First, international female Master's students may be unfamiliar with the required behaviors for conducting research or pursuing an academic career in CS [2, 10]. Second, the Master's students may have little interest in seeking out research opportunities and may not value CS research [8, 38]. Third, the students may lack confidence in their ability to succeed at conducting CS research, which may result in their avoiding research opportunities [2, 37]. Finally, international female Master's students may feel concern about belonging in academic CS and, in turn, avoid these potentially threatening environments [25, 43].

Fortunately, subtle changes in the classroom environment can influence self-concepts and possible future selves and help address these barriers [23, 47]. One such fairly minor but beneficial change to the CS graduate classroom is adding PhD student peer assistants, who may act as role models for the Master's students and

encourage their interest in PhD degrees and academic careers. There are multiple theoretical frameworks outlining the benefits of role models. As one example, the Motivational Theory of Role Modeling identifies three specific functions of role models—as behavioral models, as inspiration, and as representations of the possible [24]. Role models can act as behavior models by displaying the correct actions and behaviors that are necessary to achieve the goal (i.e., being an academic computer scientist; [10, 19]). Thus, the peer assistant role models, who are successful PhD students, demonstrate how to be productive computer scientist researchers and provide advice about conducting research and entering PhD programs. Beyond acting as behavioral models, because individuals aspire to be like their role models, role models act as inspiration to encourage individuals to value certain domains and be attracted to those fields [11, 31]. Indeed, researchers have found that the more that high school students value science courses (e.g., see them as fun and enjoyable), the more likely these students take part in after-school science activities and report intentions to pursue science careers [26]. The peer assistant role models therefore may encourage Master’s students to see academic CS as a desirable pursuit and promote the students’ interest in completing research theses or earning a PhD. Finally, role models can show individuals that it is possible to be successful in a given domain and encourage self-efficacy, which is critical for promoting students’ motivation and engagement [17, 26]. Consequently, successful PhD student role models will show CS Master’s students that it is possible to flourish in academic CS and help Master’s students feel confident in their ability to conduct CS research.

Morgenroth et al.’ [24] theory of role models nicely aligns with Dasgupta’s [6] Stereotype Inoculation Model, which posits that role models help “inoculate” students against harmful beliefs about their group in potentially threatening domains (e.g., women in STEM fields). According to this theoretical framework, when the Master’s students identify with the peer assistant role models, the Master’s students will see their future selves as academic CS researchers, which will in turn increase their interest and self-efficacy in CS research [6]. Adding to the Motivational Theory of Role Modeling, the Stereotype Inoculation Model further asserts that role models can encourage belonging and fit in certain fields [47]. By inoculating against threatening beliefs about one’s group in a specific environment, a role model indicates that one’s identity will be valued and encourages belonging in that environment. Although international female Master’s students may be less negatively impacted by masculine stereotypes about STEM than US women, these female Master’s students nevertheless may believe their place is in industry, rather than in academic CS. Thus, a peer assistant role model can dispel this belief and encourage in belonging in a CS PhD program. Taken together, both the Motivational Theory of Role Modeling and the Stereotype Inoculation Model suggest that interacting with a PhD peer assistant role model will help female Master’s students view themselves as CS researchers. Supporting this possibility, past work has found that exposure to successful female scientists and professors increased female STEM undergraduate majors’ identification with the sciences (i.e., changed their self-concept) and aspirations to pursue a career in STEM (i.e., altered their possible future self) and enhanced their sense of belonging in STEM [47]. Additionally,

previous work has found that advanced female students can act as peer role models for younger female students and encourage their confidence, belonging, and interest in STEM [7]. Although there are empirically documented benefits associated with brief exposure to role models [47], frequent and quality contact with role models can have a stronger and longer-lasting impact on students' identification with and career aspirations in a given field [1]. Master's students will have many opportunities to interact with the PhD student peer assistant role models, and these interactions will be of high quality, and hence, this may be the ideal situation to expose Master's students to role models in CS.

It is important to note certain characteristics make role models more or less effective. For instance, it is imperative that women feel similar to the role model for the role model to inspire changes in self-concepts and possible future selves [4, 18, 34]. Students generally feel more similar to and, in turn, are more inspired by role models with a matching in-group identity [6]. As one example, female students report higher interest in STEM careers after interacting with a female rather than a male scientist/potential role model [7, 47]. Because both the Master's students and the PhD student peer assistants will be students in CS (i.e., part of the graduate student in CS in-group), we anticipate that the Master's students will feel similar to the PhD students and the PhD students will be beneficial for all Master's students. Additionally, we recruited a high percentage of female peer assistants to ensure that the majority of the female Master's students had the opportunity to work with successful female role model. We anticipated that the peer assistant role models would encourage Master's students (a) to develop a CS research identity/self-concept, (b) to value CS research, (c) to feel self-efficacious in conducting CS research, (d) to have a sense of belonging in academic CS, and (e) to indicate an interest in pursuing a career in academic CS.

1.1 Aims and Objectives of the Paper

The current research reported in this paper aims to develop and test a new technique to recruit terminal Master's students into computer science (CS) PhD programs and academic CS in order to increase diversity in academic CS. We introduce peer assistants to CSCI 549: Intelligent Systems, a popular course for Master's students, which has in-class group research projects (worth 40% of the final grade). These peer assistants are successful PhD student researchers, who assist the Master's students with the in-class projects. Our hope was that the PhD students would act as role models to promote the Master's students' identification with and valuing and self-efficacy of CS research and belonging and interest in academic CS. Many of the Master's students enrolled in Intelligent Systems are international female students, and thus, there is considerable diversity in this course. However, many of these women earn a terminal Master's degree and do not continue on to earn a PhD in CS. These international female Master's students, therefore, represent an untapped

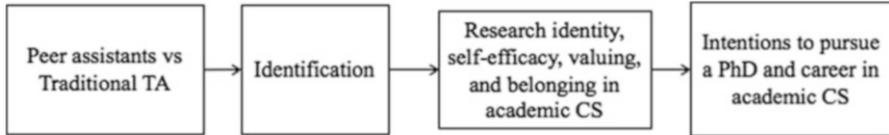


Fig. 1 Proposed theoretical model for how peer assistant role models can help broaden participation in CS

potential for enhancing diversity in academic CS. The current paper has two larger aims with related subgoals:

1. *To improve a popular CS course (CSCI 549: Intelligent Systems) for Master's students and encourage positive perceptions of CS research:* To examine whether PhD student peer assistants act as role models to encourage Master's students' research identity/self-concept, valuing, and self-efficacy of CS research, belonging in academic CS, and interest in academic CS
2. *To increase diversity in academic CS:* To examine benefits of PhD peer assistants specifically in a class with a high percentage of female and international students

Because role models are most inspirational when individuals feel similar to the role models, a secondary aim was to explore whether feeling similar to a successful PhD student would relate to more positive outcomes (i.e., enhanced research identity, self-efficacy, valuing, belonging, and interest in academic CS). We specifically compared students' identification with their teaching assistant in the control class (i.e., a successful PhD student, with whom Master's students had little interaction) to students' identification with the peer assistants in the intervention course. We also expected that a higher research identity/self-concept and more valuing, self-efficacy, and belonging in CS would relate to a higher interest in academic CS. Thus, the paper also has the following additional aim with accompanying subgoals (see Fig. 1):

3. *Add to research on importance of role models for Master's students*
 - (a) To explore whether felt similarity with the PhD student peer assistants relates to positive outcomes (i.e., enhanced research identity, self-efficacy, valuing, belonging, and interest in academic CS)
 - (b) To test whether higher research identity and more self-efficacy, valuing, and belonging in CS relate to a higher interest in academic CS

In the intervention class only, we included a variety of exploratory measures assessing perceptions of the peer assistants to examine what factors might make these PhD students more or less successful role models. Past work has found that scientists function as more effective role models when students feel like the scientists care about them [16] or when the scientists have overcome past struggles [32]. Thus, we examined whether perceiving the peer assistants as caring about the students and as overcoming similar past challenges as the students related

to enhanced research identity, valuing, self-efficacy, belonging, and interest in academic CS. We also assessed how often students met with the peer assistants and whether the peer assistants provided useful advice. Consequently, we had the following exploratory goal:

4. *Explore what factors may make peer assistants more or less effective role models.*

Participants and Recruitment Participants for the proposed research were students enrolled CSCI 54900 (Intelligent Systems). Co-author Mukhopadhyay was the instructor of this course and, hence, recruited the Master's students for this research. In particular, the Master's students received surveys from the research team via email, and co-author Mukhopadhyay, with institutional IRB approval, encouraged students to complete these assessments.

2 Assessment and Evaluation Plan

As previously mentioned, the peer assistant role model intervention was implemented in CSCI 54900 (Intelligent Systems). This class is a graduate-level course in CS, offered each year in the fall semester with approximately 40 students enrolled in it every year (about 60% of whom are women, and 90% of whom are international students). The official course description is:

This course will discuss problems in the area of intelligent systems. Topics include the formalisms within which these problems are studied, the computational methods that have been proposed for their solution, and the real-world technological systems to which these methods have been applied.

Co-author Mukhopadhyay designed this course in 1994 when he joined IUPUI and has since taught it every year over the past 25 years. Forty percent of the course grade is determined by in-class research projects, requiring research work, oral project presentations, and written project reports. Forty percent of the course is determined by a midterm exam, and this is the only assignment students do individually. In-class research projects are group effort with each group consisting of randomly assigned five or six students. As evident from the student course evaluations, students typically enjoy the hands-on research project component of the course, with a sample student comment being “the format of the class is great, as it lets students work on a project of their choice for the most part of the semester.”

Comparison Control Course In *fall 2018*, CSCI 54900 was taught in its usual style (without peer assistant role models) for baseline data collection. Pre- and post- survey instruments, as designed by co-authors Pietri and Ashburn-Nardo, were used to assess the students' research identity, value of, and self-efficacy in research, perceived belonging and fit in academic CS, and interest in academic CS (see Sect. 2.1). We also examined students' identification with the control course TA, who was a White male PhD student. The assessments from this course were used as the control comparison for the intervention peer assistant course, and this research had

a quasi-experimental design. Ideally, the control course would be taught at the same semester as the intervention course; however, this design is not feasible because of the limited number of graduate students enrolled in CSCI 54900 during a fall semester. Nevertheless, many factors were held constant, including the instructor and the general format and content of the course. The only difference between the intervention and the control course was the implementation of peer assistant role models. Similar comparisons (i.e., across semester comparisons) have been implemented in past STEM education research (see [15]).

Identification of Peer Assistant Role Models In the *spring and summer of 2019*, co-author Mukhopadhyay identified potential peer assistants from the more senior students (i.e., successful PhD students). The identified peer assistants underwent a brief training on how best to help the Master's students on their group projects in CSCI 54900 while also discussing the benefits of pursuing CS research. There were two male and four female peer assistants in the intervention course.

Intervention Course In the *fall of 2019*, the selected and trained peer assistant role models were implemented in CSCI 54900. The course had the same format as the control course, in that students again work in groups of seven or eight on a research project that makes up 40% of their final grade. The only change in the intervention course was the addition of the peer assistant role models, with one peer assistant being assigned to each project group. The peer assistant provided guidance and advice as the Master's students developed their research projects. Importantly, to ensure the students still benefited from the group project and active learning environment, the peer assistants were trained to avoid taking over the project or telling the Master's what to do for the project. The peer assistants however discussed how the project was indicative of CS research in a PhD program and related the project back to their own research. Master's students in the course completed the same pre- and post-assessments as the control course.

2.1 Evaluation Instruments

All evaluation instruments were completed by the students online using a secure survey software (i.e., Qualtrics).

Measures Completed by Students in Both the Control and Intervention Class

For all of our outcome measures, we calculated the means of the items, with higher scores indicating more of the measured construct. Our primary measure of interest was *intentions to pursue an academic CS career*. This was a two-item measure, in which students rated how likely (1, *Not at all likely*, to 7, *Extremely likely*) they were to “pursue a PhD in computer science” and “pursue a career in academic computer science.” Across both the control and intervention class, we assessed participants' identification with successful PhD student and potential role model. Specifically, in the control class, we assessed identification with the teaching assistant (TA),

and in the intervention class, we examined identification with their assigned peer assistant (PA). To index identification, participants rated their agreement (1, *Strongly disagree*, to 7, *Strongly agree*) with seven statements from the *self-other overlap measure*, e.g., “To what extent do you feel similar to your peer assistant [the teaching assistant]?” (items from [13]). Students also rated their agreement (1, *Strongly disagree*, to 5, *Strongly agree*) with four items to examine *their computer scientist researcher identity* (e.g., “Being a computer scientist researcher is an important part of my self-image”; items from [40]).

To explore students’ *valuing and self-efficacy in conducting computer science research*, students rated their agreement with the ten items in the effective motivation in computer science subscale (e.g., “I like working on computer science research”) and the ten items (1, *Strongly disagree*, to 5, *Strongly agree*) in the confidence in learning computer science subscale (e.g., “I am sure I can do advance work and research in computer science”) from the computer science attitudes scale [46]. This scale has been successfully employed in previous educational research in computer science (e.g., [21, 45]).

Finally, students completed two measures assessing their belonging and fit in CS PhD programs. Specifically participants rated their agreement (1, *Strongly disagree*, to 5, *Strongly agree*) with eight items indexing their *belonging in academic CS* (e.g., “I belong in computer science”; “I can be myself in a computer science PhD program”; three items taken from [42]; five items taken from [14]; fully eight items used in Pietri et al. [33, 34]) and with five items examining *trust and comfort in academic CS* (e.g., “I think I could ‘be myself’ in a computer science PhD program”; items from [36]).

Measures Completed by Students in Peer Assistant Intervention Course

We also included a series of exploratory measures to assess what factors may make peer assistants more or less effective. All of the following measures were completed at the end of the semester (i.e., time 2) in the intervention class. We measured how often students meet with the peer leaders (1 = “0,” 2 = “1–2,” 3 = “3–4,” 4 = “5–6,” 5 = “7–8,” 6 = “9–10,” 7 = “11–12,” 8 = “13–14,” 9 = “15–16,” 10 = “17 or more”) both virtually and in-person. Participants also rated their agreement (1, *Strongly disagree*, to 7, *Strongly agree*) with two items measuring their perception *their peer assistant cared about helping them* (i.e., “My peer assistant cared about helping me succeed in the course”; “My peer assistant cared about helping me succeed in computer science generally”), three items examining perceptions that *their peer assistant provided useful advice* (i.e., “My peer assistant provided advice about the course”; “My peer assistant provided advice about how to be successful in computer science generally”; “My peer assistant provided advice about computer science research”), and two items assessing beliefs that *the peer assistant had overcome similar past challenges* as the students (i.e., “My peer assistant and I have had to overcome similar struggles”; “My peer assistant discussed some of his/her/their past challenges in computer science”).

2.2 Overall Anticipations from the Evaluation Plan

1. Peer assistants will improve Master's students' computer scientist researcher self-concept (i.e., identification with CS research), value of CS research, self-efficacy with CS research, sense of belonging and fit in academic CS, and, importantly, interest in a career in academic CS.
2. Enhancing Master's students' computer scientist researcher identity/self-concept, value of CS research, self-efficacy with CS research, and sense of belonging and fit in academic CS will relate to a desire to pursue a career in academic CS.
3. The more the Master's students identify with the teaching assistant/peer assistants, the greater their researcher self-concept, value of CS research, self-efficacy with CS research, and sense of belonging and fit in academic CS.

3 Results of Evaluation and Assessment

We had 28 students in the control class (18 men, 10 women; 3 White, 2 East Asian, 19 South Asian, 2 Southeast Asian, 1 Other; 4 born in the USA and 24 born outside of the USA) and 44 students in the intervention class (31 men, 13 women; 6 White, 2 Black/African American, 1 Latin, 10 East Asian, 18 South Asian, 4 Southeast Asian, 3 Other; 5 born in the USA and 39 born outside of the USA). Full results are presented in Tables 1, 2, 3, and 4 included on pages 9 and 10 of this paper.

3.1 Analyses and Conclusions

Students identified more strongly with the peer assistants in the intervention course than the teaching assistant in the control courses, and this effect was consistent across both time points (see Table 1). There also was a tendency for students in

Table 1 ANOVAs predicting all outcomes from Class (intervention vs. control) and time

Measure	Class		Time		Class X time	
	<i>F</i> -value	<i>p</i> -value	<i>F</i> -value	<i>p</i> -value	<i>F</i> -value	<i>p</i> -value
Self-other overlap with TA/PA	14.51	<0.001***	0.21	0.647	1.29	0.260
Interest in academic career	0.90	0.347	4.25	0.043*	0.46	0.498
Research identity	3.07	0.084 ^a	0.27	0.606	<0.001	0.997
Self-efficacy	0.32	0.573	0.07	0.789	0.95	0.335
Value	0.05	0.819	0.20	0.654	0.58	0.449
Belonging	0.17	0.679	0.003	0.958	0.14	0.712
Trust and comfort	1.02	0.319	0.01	0.930	0.64	0.428

the intervention class to report a higher CS research identity/self-concept than those in the control class (see Table 1). For our primary outcome, interest in academic CS, there was a significant effect of time, such that students reported a higher interest in academic CS at time 2 (i.e., at the end of the semester) than at time 1 (i.e., at the beginning of the semester; see Table 1). However, when we look more closely at this finding across both classes, we see that there was only a significant increase in interest in academic CS in the intervention class (and not in the control class; see Table 2). There were no significant effects of time or intervention versus control class on self-efficacy, valuing, belonging, and trust and comfort (see Tables 1 and 2).

Table 2 Mean, standard deviations, reliabilities across time and semesters, and changes from time 1 to time 2

Measure	Time 1 <i>M</i> (<i>SD</i>)	Time 1 reliability	Time 2 <i>M</i> (<i>SD</i>)	Time 2 reliability	Mean difference (Time 2 -Time2)	<i>p</i> -value for difference
<i>Fall 18 (control) class</i>						
Self-other overlap with TA	3.95 (1.06)	$\alpha = 0.94$	4.25 (1.05)	$\alpha = 0.96$	0.30	0.318
Interest in academic career	3.96 (1.68)	$r = 0.40$	4.17 (1.54)	$r = 0.38$	0.21	0.387
Research identity	3.15 (0.93)	$\alpha = 0.80$	3.10 (1.12)	$\alpha = 0.93$	-0.06	0.743
Self-efficacy	3.79 (0.94)	$\alpha = 0.94$	3.65 (0.93)	$\alpha = 0.95$	-0.05	0.659
Value	4.00 (0.61)	$\alpha = 0.82$	3.92 (0.68)	$\alpha = 0.86$	-0.09	0.447
Belonging	3.63 (0.64)	$\alpha = 0.78$	3.61 (0.74)	$\alpha = 0.84$	-0.02	0.842
Trust and comfort	4.09 (0.67)	$\alpha = 0.82$	4.02 (0.82)	$\alpha = 0.85$	-0.07	0.656
<i>Fall 19 (intervention) class</i>						
Self-other overlap with PA	4.95 (1.11)	$\alpha = 0.97$	4.82 (1.23)	$\alpha = 0.97$	-0.13	0.581
Interest in academic career	4.19 (1.58)	$r = 0.33$	4.61 (1.48)	$r = 0.40$	0.42	0.028*
Research Identity	3.53 (0.98)	$\alpha = 0.89$	3.47 (0.89)	$\alpha = 0.81$	-0.06	0.674
Self-efficacy	3.74 (0.82)	$\alpha = 0.92$	3.83 (0.80)	$\alpha = 0.93$	0.09	0.313
Value	3.92 (0.65)	$\alpha = 0.85$	3.95 (0.60)	$\alpha = 0.85$	0.02	0.800
Belonging	3.54 (0.64)	$\alpha = 0.78$	3.56 (0.66)	$\alpha = 0.80$	0.03	0.730
Trust and Comfort	3.86 (0.77)	$\alpha = 0.83$	3.95 (0.64)	$\alpha = 0.80$	0.09	0.470

Identifying (i.e., self-other overlap) with the teaching assistant/peer assistants (TA/PA) at time 2 related to stronger CS research identity/self-concept and interest in academic CS career at time 2 and marginally correlated with higher belonging at time 2 (see Table 3 for correlations). However, contrary to predictions, identifying with the TA/PA did not relate to self-efficacy, valuing, or trust and comfort (see Table 3 for correlations). In line with our predictions, research identity/self-concept, self-efficacy, and valuing of CS research, belonging, and trust and comfort in academic CS all related to a stronger desire to pursue a career in academic CS (see Table 3 for correlations).

With regard to our exploratory measures, we found that the number of times students met with the peer assistant (virtually or in-person) did not relate to any of the outcomes at time 2. Believing the peer assistant cared about helping the students related to more identification with the peer assistant and more belonging in academic CS. Feeling like the peer assistant provided advice related significantly to higher identification with the peer assistant and marginally higher interest in academic CS and belonging in academic CS. Finally, perceiving the peer leader has faced similar past struggles related to significantly more identification with the peer assistant and trust and comfort in academic CS and marginally more interest in academic CS (see Table 4 for correlations).

Taken together, the current findings suggest multiple benefits associated with incorporating peer assistants in a Master's level CS courses. Generally, students identified more with the peer assistants than with a traditional TA. Feeling similar to a successful PhD student and potential role model related to having a higher research identity self-concept, which in turn related to a stronger interest in academic CS. Consequently, we found that students in the intervention class tended to have a higher research identity than those in the control class. Moreover, students in the intervention class showed an enhanced interest in pursuing an academic CS career from the start of the semester to the end of the semester, whereas students in the control class did not have this same increase.

Although there were positive outcomes associated with the intervention class, we did not find all of the predicted benefits. That is, we did not see any effects of time or intervention class on self-efficacy and value of CS research or belonging and trust and comfort in academic CS. Our analyses demonstrated that these are important constructs because valuing, self-efficacy, belonging, and trust and comfort all significantly related to higher interest in academic CS. Thus, it will be important to modify the peer assistant intervention to help enhance these outcomes. For instance, our exploratory assessments suggest actions peer assistants might take to positively impact belonging and trust and comfort in academic CS. That is, peer assistants should clearly communicate they care about helping the students and also discuss past struggles they have faced and overcome in CS research. Indeed, past work has demonstrated that the more students perceive STEM instructors and role models as caring about their success, the more belonging they feel in STEM [5, 16]. Additional work has found that believing a scientist has persevered in the face of past challenges encourages identification with that scientist [32]. Talking about overcoming struggles and difficulties in CS research also might promote a growth

Table 3 Correlations across outcome variables at time 1 and 2

Measure	1	2	3	4	5	6	7	8	9	10	11	12	13	15
1. Overlap with TA/PA T1.	-													
2. Overlap with TA/PA T2	0.16	-												
3. Interest in academic career T1	0.15	0.16	-											
4. Interest in academic career T2	0.08	0.27*	0.69***	-										
5. Research identity T1	0.20 ⁺	0.12	0.37**	0.34**	-									
6. Research identity T2	0.10	0.24*	0.24*	0.42***	0.59***	-								
7. Self-efficacy T1	0.07	0.02	0.59***	0.51***	0.33**	0.33**	-							
8. Self-efficacy T2	0.06	0.14	0.53***	0.65***	0.29**	0.43***	0.75***	-						
9. Value T1	0.09	-0.02	0.52***	0.38***	0.22 ⁺	0.25*	0.77***	0.53***	-					
10. Value T2	0.08	0.02	0.47***	0.61***	0.23 ⁺	0.38**	0.66***	0.86***	0.51***	-				
11. Belonging T1.	-0.01	0.11	0.45***	0.43***	0.29*	0.38**	0.65***	0.48***	0.55***	0.43***	-			
12. Belonging T2	0.08	0.22 ⁺	0.59***	0.67***	0.24*	0.39**	0.69***	0.72***	0.55***	0.71***	0.75***	-		
13. Trust and comfort T1.	-0.04	-0.07	0.44***	0.34*	0.30**	0.35**	0.65***	0.52***	0.60***	0.45***	0.70***	0.59***	-	
14. Trust and comfort T2	0.08	0.14	0.38***	0.50***	0.20 ⁺	0.29*	0.43***	0.60***	0.32**	0.60***	0.44**	0.70***	0.41***	-

$p < .05$, ** $p < .01$, *** $p < .001$

Table 4 Means, standard deviations, reliabilities, and correlations for exploratory measures in the intervention course time 2

	M (SD)	Reliability 1	Overlap with PA T2	Interest in research career T2	Research identity T2	Self-efficacy T2	Value T2	Belonging T2	Trust and comfort T2
Meeting in person	5.64 (1.73)	N/A	0.13	-0.08	0.07	0.09	0.20	0.05	0.05
Meeting virtually	2.84 (2.24)	N/A	-0.004	0.15	0.14	-0.09	-0.05	-0.001	-0.07
Peer assistant cares	5.67 (1.11)	$r = 0.79$	0.54 ^{***}	0.17	-0.15	0.08	0.10	0.30 [*]	0.19
Peer assistant advice	5.52 (1.29)	$\alpha = 0.96$	0.51 ^{***}	0.26 ^a	-0.09	0.08	0.08	0.26 ^a	0.19
Peer assistant past struggles	4.92 (1.23)	$r = 0.68$	0.56 ^{***}	0.27 ^a	-0.06	0.22	0.09	0.23	0.34 [*]

^a $p < 0.10$, ^{*} $p < 0.05$, ^{**} $p < 0.01$, ^{***} $p < 0.001$

mindset (i.e., the belief that students can and will improve at CS research), which also has been linked to higher belonging in science classes [14]. In future versions of the peer assistant intervention, it will be important to train the peer assistants to communicate caring about their students and to discuss past challenges they have faced and overcome in CS research.

Because of our small sample size, we were not significantly powered to explore differences between male and female students; however, this will be an important question for future research. Moreover, in the current work, we could not test whether certain peer assistants were more or less effective for female students. That is, female students may benefit most from peer assistants matching their gender [47] or matching their race and their gender [16, 32-34]. Having a peer assistant who matches multiple identities and who is caring and discusses past struggles may function as a particular effective intervention to spark Master's attraction to academic CS. Exploring this possibility will require implementing the peer assistant classroom intervention across multiple Master's courses to systematically test for these differences. Moreover, future work also will need to test whether this intervention is effective across other Master's level courses. Although a larger-scale study will be a critical next step in testing this intervention, we found initial evidence that peer assistants can act as inspiring role models to encourage Master's students' research identity/self-concept and interest in academic CS. When employed in Master's courses with a high percentage of female and international students, peer assistants have the potential to diversify academic CS and help address the national CS workforce need.

Acknowledgments This study has been supported by an IUPUI STEM Education Innovation and Research Institute (SEIRI) SSG grant.

References

1. S. Asgari, N. Dasgupta, N.G. Cote, When does contact with successful ingroup members change self-stereotypes? *Soc. Psychol.* **41**, 203–211 (2010). <https://doi.org/10.1027/1864-9335/a000028>
2. A. Bandura, Self-efficacy: Toward a unifying theory of behavioral change. *Psychol. Rev.* **84**, 191–215 (1977). <https://doi.org/10.1037/0033-295X.84.2.191>
3. Bureau of Labor (2014). *Statistics Projections*. Retrieved from <https://www.bls.gov/oes/tables.htm>
4. S. Cheryan, V.C. Plaut, C. Handron, L. Hudson, The stereotypical computer scientist: Gendered media representations as a barrier to inclusion for women. *Sex Roles* **69**(1–2), 58–71 (2013). <https://doi.org/10.1007/s11199-013-0296-x>
5. B.L. Christe, The importance of faculty-student connections in STEM disciplines. *J. STEM Educ.: Innov. Res.* **14**, 23–26 (2013)
6. N. Dasgupta, Ingroup experts and peers as social vaccines who inoculate the self-concept: The stereotype inoculation model. *Psychol. Inq.* **22**, 231–246 (2011). <https://doi.org/10.1080/1047840X.2011.607313>
7. T.C. Dennehy, N. Dasgupta, Female peer mentors early in college increase women's positive academic experiences and retention in engineering. *Proc. Natl. Acad. Sci.* **114**, 5964–5969 (2017)

8. J. Eccles, Expectancies, values, and academic behaviors, in *Achievement and Achievement Motivation*, ed. by J. T. Spence, (Freeman, San Francisco, 1983), pp. 75–146
9. R.B. Freeman, W. Huang, Collaborating with people like me: Ethnic coauthorship within the United States. *J. Labor Econ.* **33**, S289–S318 (2015)
10. Ibarra, H., & Petriglieri, J. L. (2008). Impossible Selves: Image Strategies and Identity Threat in Professional Women’s Career Transitions (INSEAD Working Paper). Retrieved from the INSEAD website: http://www.insead.edu/facultyresearch/research/details_papers.cfm?id18683
11. D. Gauntlett, *Media, Gender, and Identity* (Routledge, London, 2002). <https://doi.org/10.4324/9780203360798>
12. D.E. Gibson, Role models in career development: New directions for theory and research. *J. Vocat. Behav.* **65**, 134–156 (2004)
13. N.J. Goldstein, I.S. Vezech, J.R. Shapiro, Perceived perspective taking: When others walk in our shoes. *J. Pers. Soc. Psychol.* **106**(6), 941–960 (2014). <https://doi.org/10.1037/a0036395>
14. C. Good, A. Rattan, C.S. Dweck, Why do women opt out? Sense of belonging and women’s representation in mathematics. *J. Pers. Soc. Psychol.* **102**, 700–717 (2012). <https://doi.org/10.1037/a0026659>
15. D. Gross, E.S. Pietri, G. Anderson, K. Moyano-Camihort, M.J. Graham, Increased preclass preparation underlies student outcome improvement in the flipped classroom. *CBE-Life Sci. Educ.* **14**, ar36 (2015)
16. I.R. Johnson, E.S. Pietri, F. Fullilove, S. Mowrer, Exploring identity-safety cues and allyship among Black women students in STEM environments. *Psychol. Women Q.* **43**, 131–150 (2019). <https://doi.org/10.1177/0361684319830926>
17. P. Lockwood, “Someone like me can be successful”: Do college students need same-gender role models? *Psychol. Women Q.* **30**, 36–46 (2006). <https://doi.org/10.1111/j.1471-6402.2006.00260.x>
18. P. Lockwood, Z. Kunda, Increasing the salience of one’s best selves can undermine inspiration by outstanding role models. *J. Pers. Soc. Psychol.* **76**, 214–228 (1999). <https://doi.org/10.1037/0022-3514.76.2.214>
19. T.D. Kemper, Reference groups, socialization and achievement. *Am. Sociol. Rev.* **33**, 31–45 (1968). <https://doi.org/10.2307/2092238>
20. H.R. Markus, S. Kitayama, Culture and the self: Implications for cognition, emotion, and motivation. *Psychol. Rev.* **98**, 224–253 (1991). <https://doi.org/10.1037/0033-295X.98.2.224>
21. B. Magerko, J. Freeman, T. McKlin, S. McCoid, T. Jenkins, E. Livingston, Tackling engagement in computing with computational music remixing, in *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*, (ACM, 2013), pp. 657–662
22. H.R. Markus, S. Kitayama, Cultures and selves: A cycle of mutual constitution. *Perspect. Psychol. Sci.* **5**, 420–430 (2010). <https://doi.org/10.1177/1745691610375557>
23. H. Markus, P. Nurius, Possible selves. *Am. Psychol.* **41**, 954–969 (1986). <https://doi.org/10.1037/0003-066X.41.9.954>
24. T. Morgenroth, M.K. Ryan, K. Peters, The motivational theory of role modeling: How role models influence role aspirants’ goals. *Rev. Gen. Psychol.* **19**, 465–483 (2015)
25. M.C. Murphy, C.M. Steele, J.J. Gross, Signaling threat how situational cues affect women in math, science, and engineering settings. *Psychol. Sci.* **18**, 879–885 (2007). <https://doi.org/10.1111/j.1467-9280.2007.01995.x>
26. B. Nagengast, H.W. Marsh, L.F. Scalas, M.K. Xu, K.T. Hau, U. Trautwein, Who took the “x” out of expectancy–value theory? A psychological mystery, a substantive-methodological synergy, and a cross-national generalization. *Psychol. Sci.* **22**, 1058–1066 (2011). <https://doi.org/10.1177/0956797611415540>
27. National Science Board, *Science and Engineering Indicators 2016* (National Science Foundation, NSB-2016-1, 2016, 2016)
28. National Science Foundation, National Center for Science and Engineering Statistics (2015). Women, Minorities, and Persons with Disabilities in Science and Engineering: 2015.. Special Report NSF 15-311. Retrieved from <http://www.nsf.gov/statistics/wmpd/>

29. B.A. Nosek, F.L. Smyth, J.J. Hansen, T. Devos, N.M. Lindner, K.A. Ranganath, et al., Pervasiveness and correlates of implicit attitudes and stereotypes. *Eur. Rev. Soc. Psychol.* **18**, 36–88 (2007). <https://doi.org/10.1080/10463280701489053>
30. S.E. Page, Making the difference: Applying a logic of diversity. *Acad. Manag. Perspect.* **21**, 6–20 (2007)
31. E. Paice, S. Heard, F. Moss, How important are role models in making good doctors? *BMJ: Br. Med. J.* **325**, 707–710 (2002). <https://doi.org/10.1136/bmj.325.7366.707>
32. E.S. Pietri, M.L. Drawbaugh, A.N. Lewis, I.R. Johnson, Who encourages Latina women to feel a sense of identity-safety in STEM? *J. Exp. Soc. Psychol.* **84** (2019). <https://doi.org/10.1016/j.jesp.2019.103827>
33. E.S. Pietri, I.R. Johnson, E. Ozgumus, One size may not fit all: Exploring how the intersection of race and gender and stigma consciousness predict effective identity-safe cues for Black women. *J. Exp. Soc. Psychol.* **74**, 291–306 (2018a)
34. E.S. Pietri, I.R. Johnson, E. Ozgumus, A.I. Young, Maybe she is relatable: Increasing women’s awareness of gender bias encourages their identification with women scientists. *Psychol. Women Q.* **42**, 192–219 (2018b). <https://doi.org/10.1177/0361684317752643>
35. President’s Council of Advisors on Science and Technology. Engage to excel: Producing one million additional college graduates with degrees in science, technology, engineering, and mathematics(2012). , Retrieved from http://www.whitehouse.gov/sites/default/files/microsites/ostp/pcast-executive-reportfinal_2-13-12.pdf
36. V. Purdie-Vaughns, C.M. Steele, P.G. Davies, R. Diltmann, J.R. Crosby, Social identity contingencies: How diversity cues signal threat or safety for African Americans in mainstream institutions. *J. Pers. Soc. Psychol.* **94**, 615–630 (2008). <https://doi.org/10.1037/0022-3514.94.4.615>
37. R.M. Ryan, E.L. Deci, Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being. *Am. Psychol.* **55**, 68–78 (2000)
38. U. Schiefele, Interest and learning from text. *Sci. Stud. Read.* **3**, 257–279 (1999)
39. T. Schmader, B. Major, The impact of ingroup vs outgroup performance on personal values. *J. Exp. Soc. Psychol.* **35**, 47–67 (1999). <https://doi.org/10.1006/jesp.1998.1372>
40. I.H. Settles, When multiple identities interfere: The role of identity centrality. *Personal. Soc. Psychol. Bull.* **30**, 487–500 (2004). <https://doi.org/10.1177/0146167203261885>
41. L.R. Tropp, S.C. Wright, Ingroup identification as the inclusion of ingroup in the self. *Personal. Soc. Psychol. Bull.* **27**, 585–600 (2001). <https://doi.org/10.1177/0146167201275007>
42. G.M. Walton, G.L. Cohen, A question of belonging: Race, social fit, and achievement. *J. Pers. Soc. Psychol.* **92**, 82–96 (2007). <https://doi.org/10.1037/0022-3514.92.1.82>
43. G.M. Walton, G.L. Cohen, A brief social-belonging intervention improves academic and health outcomes of minority students. *Science* **331**, 1447–1451 (2011). <https://doi.org/10.1126/science.1198364>
44. White House Briefing (2017). Expanding Access to High Quality STEM and Computer Science Education Provides More Pathways to Good Jobs. Retrieved from <https://www.whitehouse.gov/briefingsstatements/expanding-access-high-quality-stem-computer-science-education-provides-pathways-goodjobs/>
45. E. Wiebe, L. Williams, J. Petlick, N. Nagappan, S. Balik, C. Miller, M. Ferzli, Pair programming in introductory programming labs, in *Proceedings Submitted to American Society for Engineering Education Annual Conference and Exposition.* (2003)
46. L. Williams, E. Wiebe, K. Yang, M. Ferzli, C. Miller, In support of paired programming in the introductory computer science course. *Comput. Sci. Educ.* **12**, 197–212 (2002)
47. J.G. Stout, N. Dasgupta, M. Hunsinger, M.A. McManus, STEMing the tide: Using ingroup experts to inoculate women’s self-concept in science, technology, engineering, and mathematics (STEM). *J Pers Soc Psychol* **100**, 255–270 (2011)
48. C.M. Guarino, Faculty service loads and gender: are women taking care of the academic family? *Res High Educ* **23** (2017)

A Project-Based Approach to Teaching IoT



Varick L. Erickson, Pragma Varshney, and Levent Ertaul

1 Introduction

Every day, more and more everyday objects are becoming “smart.” With new capabilities such as sensors, computing power, and Internet connectivity, these smart, connected objects have significantly improved capabilities and new applications. Designing, connecting, improving, and securing these everyday “things” to the Internet is now a rapidly growing field, now known as the Internet of Things (IoT).

Since the “Internet of Things” term was first coined in 1999 [1], IoT technology has evolved and spread quickly. IoT and smart, connected devices are already impacting industrial, manufacturing, technology, healthcare, and consumer markets and will continue to do so. In 2019, \$750 billion dollars were spent on IoT [2]. By 2023, it is expected the field will grow to over one trillion dollars [3]. As this field has grown, so does the need to effectively teach and expose students to this new area of computer science.

Currently only a few institutions in the California State University system offer courses in IoT, but interest in the area is rising. As interest in IoT continues to grow, we expect the number of courses to increase as well.

One key challenge of designing and teaching IoT courses is the highly interdisciplinary nature of IoT and its increased focus on hardware. To successfully understand, design, and build IoT projects, students and educators need a broad background in software and hardware, and also need to understand how to make software and hardware work well together. In teaching and learning IoT, students and educators may be exposed to many topics outside of the traditional computer

V. L. Erickson (✉) · Pragma Varshney · Levent Ertaul
Department of Computer Science, California State University, East Bay, Long Beach, CA, USA
e-mail: varick.erickson@csueastbay.edu; pvarshney@horizon.csueastbay.edu;
levent.ertaul@csueastbay.edu

science curriculum, including hands-on hardware prototyping and debugging, wireless sensors, wireless communications, and design tradeoffs to optimize power, performance, or device size. Another important consideration is making hardware kits and tools available to students for hands-on projects and labs. Since these parts likely need to be purchased and made available to students, one must also consider budget and materials availability when designing projects and assignments. Having dedicated lab space and space for student groups to work outside of class is also very beneficial.

Even when budget, space, and materials are available, successfully selecting and integrating hardware and materials into labs can be extremely challenging and time-consuming. Many kits, hardware, and software are available, yet educators need to provide ones which are easy to use, versatile enough to be used in many projects, and reasonably priced to fit in department budgets. While IoT courses are becoming increasingly popular, very limited information is available regarding which types of hardware and kits work best with IoT courses, and how to best integrate hardware into the IoT curriculum. This lack of existing material can make setting up a new course very challenging.

With these challenges and considerations in mind, we share in the present work how we designed, prepared, and taught an IoT course at California State University, East Bay. The course requires prerequisite knowledge in data structures, networking, and C++ and was open to graduate students as well as advanced undergraduates. We have now taught the course for two semesters (Fall 2019 and Spring 2020) and are able to share lessons learned and outcomes based on these two semesters. We especially focus on sharing information about the hardware, components, and software packages used in the course and why we chose them. We anticipate that these lists of materials, lab activities, and recommendations will be beneficial to educators and others looking to develop courses in the IoT field.

We start by examining related work in Sect. 2. We examine classes that cover IoT content and the different approaches used to teach IoT at several other universities. Next, we discuss how we designed our course in Sect. 3. We then describe and list the hardware kits used in the course in Sect. 4. In Sects. 6 and 7, we discuss our first two experiences teaching the course and the lessons learned from each semester. Section 8 documents changes to the course and next steps. Finally, we summarize our findings in Sect. 9.

2 Related Work

In designing our IoT course, we first examined teaching approaches used by others in the field. As previously mentioned, IoT is a highly interdisciplinary field covering many technical areas. As a result, many different departments offer different versions of IoT. IoT courses tend to be offered under computer science, computer engineering, information systems, and electrical engineering departments [4–6]. However, course offerings have also been seen in departments such as business [7]

and even art [8]. In addition to offering individual courses, there are also programs and certificates with a focus on IoT [6].

The learning objectives, student audience, hardware vs software focus, and topics covered vary greatly from course to course. Most courses cover a combination of both software and hardware topics. For example, the instructors of [5] utilize a hardware and web-service approach. Students are given kits and work in groups of two to three students. Projects are utilized for the midterm and final exams with the topic of the project chosen by the students. A small budget is available to buy components for projects. In order to get components, students must justify the purchase to the instructor through an “investor” pitch. In addition to projects, students are also given assignments and quizzes. For their choice of platform, students utilized microcontroller hardware made by Particle [9]. An advantage of this platform is that it comes with administrative cloud software for managing devices. The drawback for these devices is that they are roughly two to four times the cost of other similar platforms.

Since hardware can be a challenging topic for software-focused students, other IoT courses choose a more application and service oriented view of IoT with little to no use of hardware. For example, the instructors of [4] use readily available REST APIs for student assignments. They start with scaffold examples showing how an existing REST API, such as a publicly available transportation API, can be used and then show how different public REST APIs can be combined. Students then create their own IoT services that can in turn be combined with other IoT services. This approach has the advantage of being able to capture the “spirit” of IoT with minimal cost, hardware, and prior knowledge. A student with basic CS knowledge would be able to understand and create IoT applications and services. However, this approach is one-dimensional. It does not capture other issues common in IoT such as sensor limitations, energy conservation, fog computing, and architecture design.

Also important to note is that group work is a common feature of many courses [4, 5]. Though not explicitly stated, this is likely due to availability of hardware. Purchasing kits, hardware components, and software licenses can get costly for both students and universities. Additionally, with components, software, and protocols frequently changing, keeping components up to date may become challenging. While group projects can be a valuable part of an IoT course, it is also important to implement peer evaluations and incentives to encourage all group members to contribute to and understand their projects [10].

3 Approach

Given the interdisciplinary and changing nature of IoT, there are many considerations needed to design a course for IoT. Do we focus on the current state of the technology and the protocols in the marketplace? Do we focus on software, hardware, or a combination of the two? If we utilize hardware, how do we handle material? How much does hardware cost? How much prior experience is needed

with hardware? Given that many different technologies are used in IoT, how do we make material deep enough to be useful yet broad enough that students do not get bogged down in unnecessary details? These are just a few of many considerations. We explain our key considerations and our thought process behind our course design in the following sections.

3.1 Audience and Assumptions

In our IoT course, we targeted the material toward graduate students and advanced undergraduate students in computer science. We needed our students to have a solid foundation in computer science. We therefore required that students have taken coursework in data structures, algorithms, and networking. We also required a familiarity or the ability to use C++ since the Arduino modules we had available in our hardware kits use C++. Since most computer science students at CSUEB have limited coursework in hardware, we assumed students have little or no exposure to embedded systems or hardware. We also limited the number of students in each class to thirty students. Keeping the class size small allowed the students to work in smaller groups of 2–3 students per group. This allowed us to stretch our materials and hardware budget to provide as many parts as possible, and helped students more easily get individualized help on projects.

3.2 Course Structure

Our goal in developing our IoT course was to provide students a broad introduction to the IoT field in general, as well as the opportunity to gain hands-on experience in hardware and prototyping for IoT. Based on our review of other IoT courses, we determined that combining hardware and software topics is essential in order for students to gain a solid introduction to IoT. Hands-on hardware projects allow students to experience firsthand some of the many challenges one might face when designing, testing, and improving IoT devices. Also, many computer science students at CSUEB would otherwise have little or no prior introduction to hardware and prototyping. The IoT course provided a valuable opportunity for these students to gain exposure to hardware. Therefore, we decided to make the course contain a significant lab component.

To allow sufficient time for both teaching IoT concepts and gaining hands-on practice, we decided that about half of the class time should be lecture-based, with the remaining half dedicated to labs and building hardware skills. The IoT course was scheduled to meet twice weekly for two 90-min periods. Each week, the first 90-min section was dedicated to lecture and introduce topics. The second 90-min section was reserved for completing hands-on lab assignments.

Given the hands-on nature of IoT, a significant portion of the course grade was dedicated to completing the hardware labs and assignments. To further emphasize the importance of students understanding each assignment, students did not simply “turn in” a finished lab. Instead, students needed to demo their working assignment with the professor or teaching assistant in order to demonstrate full understanding of the assignments. Additional office hour time was therefore reserved for “grading hours,” in which students would demonstrate that their assignment was complete and working.

Many of our students were seeking to gain experience in IoT for their resumes or for capstone projects. With this in mind, we decided to also assign a final project, allowing students to build a working IoT device. To help students apply the skills and concepts from the labs, the final project was required to embody key elements of IoT, including leveraging one or more sensors, using communication protocols, and performing data management. During week 7, instead of having a lab activity, each group of students prepared a proposal of their final project idea and presented it to the class in an “elevator pitch” presentation. During the elevator pitches, other students and the instructor would provide written feedback to help each group improve their ideas. Finally, during the last 1–2 months of the semester, students would complete the final project. Some students even chose to use their final project from the IoT course as a starting point for their capstone or thesis project.

This increased focus on labs and hardware skills also required that we design labs and hardware kits for the students. Ideally, we would be able to provide individual kits for every student; unfortunately our budget allowed us to purchase only one kit per 2–3 students. Since students would be working in groups for the entire semester, we also implemented measures to incentivize students to learn the projects and contribute equally. First, as part of the grading process, both lab partners were required to demo and explain each project. Rather than simply turning in a lab report, all group members needed to show that they understood each lab in order to receive full credit. Second, we included a peer-review component as part of the final course grade. Finally, we also included individual assessments in the course grade to ensure each student individually understood the material. The first time we offered the course, both a midterm and final exam were given. The second time we offered the course, we replaced the midterm and final exams with weekly online quizzes, to free up more class time for hands-on prototyping. Details of the grading breakdown are as follows: 30% In-Class Assignments, 5% Participation/Peer Reviews, 10% Midterm, 10% Initial/Final Project Proposals, 45% Final Project Presentation and Report.

3.3 Introducing Key Hardware Skills Through Hands-on Labs

As mentioned previously, we believe exposure to hardware is a critical component to IoT. However, since the hardware and software used in IoT can rapidly change, it can be challenging to decide which topics to focus on. We therefore chose to focus

on core concepts that are likely to persist rather than areas of IoT such as protocols, which are continually changing. One area of IoT which continues to be important is understanding how to successfully prototype and integrate hardware and software together. Rather than having students memorize protocols and hardware components which might go out of date, we kept the focus on developing key skills for designing, troubleshooting, improving, and evaluating IoT devices.

Many students in computer science are not familiar with hardware. To address this need, we specifically designed our lectures, labs, and projects to gradually introduce key hardware skills to students. These topics are shown by Table 1. We used the lecture to introduce each topic, and then the students would gain hands-on practice during the lab portion in the following class meeting.

For example, during the Sensors and Actuators week, we used the lecture portion of the class to teach students about how to read spec sheets and how sensor devices work. Then, during the lab portion, students gain firsthand practice on reading a sensor's spec sheet, connecting software to the sensor, and collecting data remotely from the sensor. Students need this experience to enable them to design feasible applications based on actual documented component performance rather than a simple notion of what each component does.

In designing and evaluating IoT projects, students also need to know how to examine specification sheets and understand hardware limitations. For example, students may not know that a gas sensor requires a 20-min warm-up period for reading to be accurate and needs to be calibrated to the current room temperature. A student wanting a wireless sensor to run off a battery will need to calculate a suitable duty cycle from a device specification sheet to achieve a target operational time. This meta-knowledge and hardware experience is not easily learned simply

Table 1 Schedule of topics covered in the IoT course

Week	Topic
1	Introduction, what is IoT
2	IoT architecture
3	Sensors and actuators
4	IEEE 802.15.4
5	Low power wide area networks
6	IoT for Homes
7	IoT management
8	IoT security
9	Fog and edge computing
10	Data analytics
11	IoT for smart cities
12	IoT for transportation
13	IoT for manufacturing
14	IoT for public safety
15	IoT for oil and gas

from lecture. By teaching IoT concepts with lecture followed by hands-on practice, we are able to provide a richer, more valuable learning experience for students.

The hands-on practice in the IoT lab also allows students to develop a different set of debugging skills. Computer science students typically only experience software errors. They rarely attribute errors they experience to the compiler, computer, or IDE as it is almost never the source of errors. As a result, many times when students experience difficulties with hardware, they often assume error are caused by their code and not the hardware. They often overlook issues such as reversed wires, improper hardware, or even broken hardware. Though not all students will need to interact with hardware at such a low level in industry, the value of debugging hardware makes students sensitive to the types of issues that hardware can cause when working on a deployment. Even when hardware is working correctly, students also need experience to debug issues caused by the environment. For example, students may not realize that a thermostat temperature sensor is deployed too close to an air vent, causing thermostat readings to be artificially high or low. Students may deploy a camera in a position such that early morning lighting conditions cause false positives for a classification application. Working with hardware in a real-life environment helps students develop intuition and debugging skills for successful hardware deployments in the field.

Though hardware is emphasized, it is not the sole focus of the course. Roughly half of the labs and topics of the class examine non-hardware concepts such as communication, security, and network architecture. These topics are shown by Table 1. While software, protocols, and architectures may change as the IoT field becomes more mature, many of the core issues such as scalability, integration, and robustness will continue to persist. By teaching students key skills to prototype, debug, integrate, and optimize their projects, we anticipate that the course will be a valuable foundation even as IoT continues to evolve.

4 Hardware

There are several key challenges utilizing a hardware-based approach to IoT. The first is supplying hardware to students. Depending on the hardware chosen, purchasing components and tools can potentially be a significant upfront cost. This cost can be partially defrayed by having students work in groups and share hardware. However, limits to sharing must also be considered as too many students sharing hardware can potentially lead to effort imbalances in groups where some students do not have the opportunity to use the hardware. For our class, we planned for two to three students per group. The second challenge is choice of hardware. There are numerous choices for the computing platform and even more for peripheral sensors and actuators. Lastly, we need to consider the toolchain and programming environment students will use. Again, there are many different approaches each with their advantages and drawbacks. In this section, we address each of these concerns.

4.1 *Microcontrollers*

Numerous choices are available for microcontrollers. There are several considerations that need to be examined to choose an appropriate platform: Availability of resources and support, overall capability, cost, and wireless communication capability. Table 2 compares several popular microcontroller options. Perhaps the most popular is the ubiquitous Arduino Uno [11], which has long been a favored microcontroller platform for educators and hobbyists. The board is moderately priced, has a large support community, and supports many electronic components through the Arduino software platform. However, this board does not have any native wireless communication and requires additional radio hardware. The board has somewhat limited computing resources. For sensing applications, this Arduino board could be sufficient, but anything requiring moderate amounts of computing may prevent the platform from being used. The platform also is not designed for low-power applications and does not have an effective deep sleep mode.

The Raspberry Pi [12] is another platform that is a popular choice. Rather than a simple microcontroller, it actually is a multicore computer that is capable of running a full Linux operating system complete with a graphical interface. This is an excellent platform where battery life is of little concern and substantial computing power is required on location. Interfacing components is well-supported and libraries exist for the most popular components. While capable, this platform is costly and is not suitable for many instances where sensing and battery life are a concern.

Particle Photon [13] boards are very capable boards designed for IoT prototyping projects. They are capable of light computing and have native Wi-Fi integrated into the board. At the time of this publication, the company offering this platform provides free cloud device management for the first 100 devices. The provided library gives a simple publisher/subscriber interface model that allows for easy data management. One downside to this platform is higher cost. The other drawback is the high energy use in deep sleep mode ($80\ \mu\text{A}$).

A similarly capable board is the NodeMCU ESP8266 [14]. This is an open-sourced design based on the ESP8266 Wi-Fi chip [15]. This design is manufactured by many different companies. While the support by these individual companies varies greatly, there is an active userbase for these devices. While not quite as capable as the other platforms, these boards are still able to do some light computing. The platform also has a very low energy deep sleep ($10\ \mu\text{A}$). Perhaps their greatest advantage is cost. In bulk, each node can be bought for \$2–\$4 at the time of this publication.

The last board we examined is the NodeMCU ESP32, which is the successor to NodeMCU ESP8266. This board has a faster processor than the NodeMCU ESP8266 and a very aggressive deep sleep. It is able to consume $2.5\ \mu\text{A}$ at a deep sleep state, which makes the platform particularly well suited for long term monitoring deployments relying on battery power. The board also has 18 analog/digital channels rather than just one analog and 17 digital channels as

Table 2 Comparison of several different microcontroller options

	Arduino Uno	Raspberry Pi 4 B		NodeMCU	ESP8266	ESP32	ESP32+LoRA	Particle photon
CPU	16 MHz	Quad Core 1.5GHz		80 MHz	240 MHz	240 MHz	240 MHz	120 MHz
SRAM	2 KB	1-4GB		16 KB	520 KB	520 KB	520 KB	128 KB
Memory	32 KB	Based on SD card		4 MB	4 MB	4 MB	4 MB	1 MB
Wireless	None	Wi-Fi, BT 5.0/BLE		Wi-Fi	Wi-Fi, BT 4.2/BLE	Wi-Fi, BT 4.2/BLE, LoRA	Wi-Fi, BT 4.2/BLE, LoRA	Wi-Fi
Standby	15 mA	600 mA		0.9 mA	0.8 mA	0.8 mA	0.8 mA	1 mA
Deep sleep	30 μ A	NA		10 μ A	10 μ A	10 μ A	10 μ A	80 μ A
GPIO	6 ADC, 14 DC	40 GPIO		1 ADC, 17 DC	18 ADC	18 ADC	18 ADC	6 ADC, 8 DC
Interfaces	1 \times UART	1 \times UART		1 \times UART	3 \times UART	3 \times UART	3 \times UART	1 \times UART
	1 \times I2C	1 \times I2C		1 \times I2C	2 \times I2C	2 \times I2C	2 \times I2C	1 \times I2C
	1 \times SPI	1 \times SPI		3 \times SPI	3 \times SPI	3 \times SPI	1 \times SPI	
Cost	\$20	\$35-\$55		\$2-\$8	\$2-\$12	\$2-\$12	\$15-\$20	\$19
Pros	Inexpensive, simple, accepts 6-20 v input	Excellent computing capability, full Linux OS with GUI, many I/O options		Fair computing capability, inexpensive, good sleep options	Very good computing capability, BT/BLE/Wi-Fi, good sleep options long inexpensive	Very good computing capability, BT/BLE/Wi-Fi, good sleep options long	Very good computing capability, BT/BLE/Wi-Fi, good sleep options long	Free cloud management Good computing capability
Cons	High energy, no wireless, limited computing capability	Very high energy, no sleep option, expensive, no long range radio		Only one analog input, no BT/BLE, no long range radio	No long range radio	No long range radio	Expensive	High energy, limited sleep options, expensive

compared with the ESP8266. There is also a slightly more expensive version of the ESP32 board made that includes a 900 MHz Long Range (LoRA) radio module that is able to communicate long distances.

For our class, we chose to use three different boards: a variation of the NodeMCU ESP32 called the M5 Stack [16] with a built-in screen and sensors, an ESP32 board with LoRA, and a Raspberry Pi (see Fig. 1). The ESP32 platforms have many input and output options, many different sleep options for long term deployments, and three different radios for various situations. The Raspberry Pi is useful for fog and edge computing as well as functioning as a simple server. This combination of platforms allows for a variety of potential IoT projects and applications at a good price point.

4.2 Sensors, Actuators, and Peripherals

Great care was taken to choose sensors, actuators, and peripherals. The goal was to minimize cost while maximizing the capability of the kit.

Sensors

When possible, we bought sensors that had multiple sensing capabilities. In particular, we chose the BME680 and APDS9960 sensors to maximize sensing with minimal sensors. The BME680 [17] is primarily a gas sensor able to detect presence of volatile organic compounds. However, it also has temperature, humidity, and barometric pressure sensors. The APDS9960 [18] is able to detect whether someone is close to the sensor using multiple light sensors. It is also able to measure light intensity and color. Using the multiple sensors, it is sensitive enough to detect simple hand gestures. By combining sensing capabilities, we are able to capture the

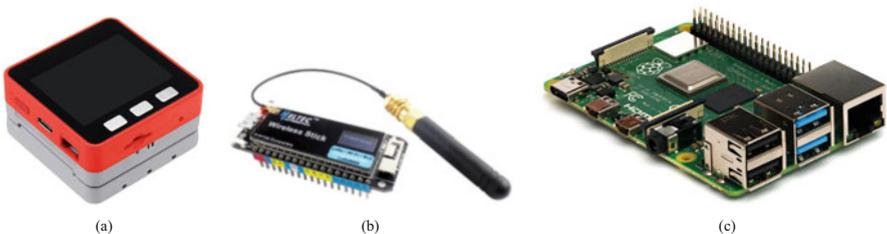


Fig. 1 The three platforms used in the class. (a) The M5Stack is a platform based on the NodeMCU ESP32. It includes, a screen, microphone, SD card reader, buttons, and LED's. (b) This is a NodeMCU ESP32 based platform that has a LoRA radio integrated into the design. (c) A Raspberry Pi 3 B+ platform was used as a server for labs and projects. Pictured is the most recently released version 4B

following with only six different sensors boards; moisture, movement, proximity, light intensity, color, hand gestures, temperature, various gases, humidity, location, and sound.

Actuators

For actuation, we include micro servos and a simple relay. As the servo is essentially a mechanical lever, it can be applied to any situation where small motion is required. For example, the servo could be used to turn on a light, push a button, or lift a latch. This makes this simple actuator very versatile. The relay can be used in situations where an electronic device needs to be turned on or off. While only two servos are included in the provided kit, we had many held back in reserve for students needing additional actuation for their final project.

Peripherals

Several peripherals were bought for the Raspberry Pi kits. The goal was to be able to provide students with a relatively portable server that could also potentially serve as a user interface for an IoT project. For this we included a 10.1” touchscreen and small keyboard to use with the Raspberry Pi kit.

4.3 Arduino vs. Real Time Operating System vs. Developer Framework

Another important consideration is the programming framework used. We considered three different options: Arduino, FreeRTOS, and the developer framework.

Arduino, which uses C++ as the programming language, is a widely used platform for microcontrollers. The platform is open-source and designed for students without an electronics background and is often used in education and by hobbyists. Code developed on one microcontroller can often be ported to another type of microcontroller without too much effort. The Arduino Software IDE is available for Windows, OSX, and Linux and is designed to easily access libraries and examples for hardware. The software also greatly simplifies the toolchain setup. One drawback is that Arduino is not a true OS and better suited for running a single application. While it is possible to create multiprocess applications with Arduino, it tends to be more complicated. Another drawback is that not all board functionality may be available through the Arduino interface.

There are many real time operating systems available (RTOS) for microcontroller platforms. FreeRTOS [19] is well known and available for the ESP32 platforms. This allows for multiple applications to be run on a platform. However, drivers are

sometimes not readily available, and writing drivers for hardware is time consuming and outside the scope of the class. Working with an RTOS may mean working with the system at a lower level of abstraction.

The last option is the provided developer framework for the ESP32 [20]. Similar to Arduino, it is primarily designed for single applications. The advantage of using the official developer framework is that all features are available. Again, hardware drivers are not always available and may be time consuming to create. This lack of hardware drivers may require working at a lower level of abstraction.

Our course focus was learning about IoT by creating applications. Thus, Arduino was a clear choice. While there may be situations where multiple applications could be useful, the vast majority of situations only require a single application running on the platform with perhaps a few simple threads.

5 Labs

5.1 Lab Preparation

Most of the lab equipment were purchased directly off the shelf and ready to use out of the box. However, some preparations were required to ensure that the component kits were ready for students. Given that students in the course had limited hardware experience, we decided to complete some soldering in advance. The following soldering was completed before the start of labs each semester:

- Solder all of the header pins onto the ESP 32 chips (Note that the headers were included with each ESP 32, but not pre-soldered).
- Solder header pins on to the PIR, gas, light, and sound sensors. (Note that the headers were included with each sensor, but not pre-soldered).

If the course were taught to students who have a stronger hardware background, this soldering could be done as part of the lab. We chose to do the soldering in advance to ensure students had components which were connected properly to avoid potential shorting or solder-related errors.

Finally, in addition to preparing the hardware kits, short lab manuals were written containing detailed instructions for each lab as well as clear deliverables (tasks the students needed to demo with the teaching assistant or instructor in order to receive credit for the lab). To encourage creativity, small amounts of extra credit were always offered for students who voluntarily chose to build on and extend the assigned project. In some of the labs, suggestions on how to extend the lab were provided. Each week's lab handouts were made available online in advance of the lab.

5.2 *Lab 1: IDE and Toolchain Setup*

During the first week of class, lab did not meet (the semester starts on a Tuesday, so there was only one lecture during the first week). Therefore, Lab 1 officially started during the second week of classes.

The focus of the first lab is to distribute the equipment and to familiarize students with the hardware and software needed for the course. First, students are instructed to form groups of two to three and each group is assigned a kit of components to use for the semester. For convenience, each kit is stored in a tool organizer with multiple compartments. Student groups needed to take home the kit and bring it back to lab week since storage space was not available.

Note that each group receives a kit of components, but students used their own laptops to install and run associated software. For the first lab, students needed to install VisualStudio Code, Platform IO, and the ESP 32 drivers. Platform IO offers the same functionality as the Arduino IDE but has other more advanced features available such as version control. The students are given instructions on how to set up Platform IO and how to program the M5Stack ESP32 to blink the LEDs and display a message on the OLED screen. In order to receive credit for the lab, students needed to demonstrate how to upload a program to M5Stack and show a successful blinking LED to the instructor or TA. This first lab serves as an effective initial checkpoint to verify that students are able to set up and connect to the hardware successfully (Tables 3 and 4).

5.3 *Lab 2: Sensors*

Lab 2 happens during the third week of the class, when sensors and actuators are introduced in lecture. In this second lab, students gain hands-on experience with sensors to build on the lecture material. Students experimented with different types of sensors (gas, gesture, and PIR sensors) along with a few different types of ESP32-based boards. The students used a breadboard to make circuits with ESP32. Using various sensors like the PIR, BME680, APDS9960 and actuator servo and relay, students then verified each sensor's functionality and observed how each component worked together. For example, by connecting the PIR along with a relay, students were able to demonstrate a PIR sensor which activated the relay whenever motion was detected.

While breadboards were provided, we found that very few students actually used the breadboards. Instead, students preferred to use jumper wires to connect directly to the sensors. However, many students would accidentally connect their sensors incorrectly. Another common error was that students would forget to read the documentation and verify that certain pins are not already dedicated to specific built-in hardware such as OLED or radio.

Table 3 This table shows the components each kit contains and the approximate cost at the time of publication

IoT Student Kit				
Component	Quantity	Description	Unit cost	Total cost
Multi-meter	1	Digital multimeter with Ohm, Volt, Amp and Diode Tester	\$11.00	\$11.00
Esp32 with LoRA	4	ESP32 with OLED screen and SX1276 LoRA transceiver module and antenna	\$20.00	\$96.00
Battery for ESP32	4	3.7 V battery with connector for ESP32	\$5.50	\$22.00
Micro servo	2	Small servo for actuation	\$2.25	\$4.50
3 V 1 channel relay power switch module	2	Relay for turning on/off devices	\$3.60	\$7.20
Breadboard	2	400 tie point solderless prototype PCB board	\$1.25	\$2.50
Soil moisture sensor	1	Capacitive soil moisture sensor corrosion resistant	\$2.00	\$2.00
Passive InfraRed (PIR)	2	OpenPIR based around the NCS36000 PIR controller	\$16.00	\$32.00
APDS9960	1	Proximity, light, RGB, and gesture sensor	\$12.00	\$12.00
BME680	2	I2C or SPI, VOC, temperature, humidity, pressure sensor	\$20.00	\$40.00
MIC sensor	1	Digital sound sensor	\$7.00	\$7.00
GPS sensor	1	GPS - 66 channel w/10 Hz updates	\$40.00	\$40.00
Breadboard wires	1	30 cm 40pin breadboard jumper wires ribbon cables kit with female to male and female to female and male to male	\$14.00	\$14.00
USB cable	4	USB 2.0 cable—A male to micro B	\$ 2.34	\$9.34
Screwdriver set	1	7 In 1 stubby multi-bit drivers pocket precision screwdriver set	\$3.65	\$3.65
Raspberry Pi 3 B+ (B Plus) starter kit	1	Raspberry Pi 3 B+ (B Plus) with 32 GB SD Card,2.5A USB power supply, and case.	\$80.00	\$80.00
10.1" Raspberry Pi 3 touchscreen	1	10" touchscreen with frame	\$126.00	\$126.00
Wireless mini keyboard with touchpad	1	Mini wireless keyboard with touchpad to use with raspberry Pi	\$15.00	\$15.00
10.1" Raspberry Pi 3 touchscreen	1	10" touchscreen with frame	\$126.00	\$126.00
Toolbox with compartments for organizing components	1	Dewalt 10-compartment pro small parts organizer or similar	\$20.00	\$20.00

Total cost: \$544.19

Table 4 The topics covered in each lab

Lab	Description	Software/libraries	Hardware
Lab 1	Distribution of kits, setting up toolchain with visual studio code and platformIO	Library: M5Stack software: platformIO [21], visual studio code [22]	M5 stack
Lab 2	Introduction to sensors and actuators	Library: SparkFun APDS9960 RGB, gesture sensor	Breadboard, ESP32, PIR, Relay, Servo, BME680, APDS9960
Lab 3	Mesh networks, embedded webserver, access points	Library: painlessMesh, ESPAsyncWebServer	ESP32, PIR, BME680, Moisture sensor
Lab 4	Bluetooth communication	Library: ESP32 BLE Arduino [23, 24], Adafruit BME680 library	ESP 32, BME 680
Lab 5	How to utilize LoRA	Library: Adafruit GPS library [25], LoRa	ESP 32, GPS sensor, soil moisture sensor
Lab 6	MQTT and COAP models	Software:MQTT.fx [26, 27] library: PubSubClient	ESP 32, BME680
Lab 7	Security	Library: Adafruit GPS library, LoRa, EduShield	ESP 32, GPS sensor, soil moisture sensor
Lab 8	Data analytics	Setting up Raspberry Pi [28], installing LAMP stack [29] installing PHPMyAdmin [30]	Raspberry Pi, ESP 32, BME680

To receive credit for this lab, students needed to demonstrate they have successfully interfaced with the sensors and are able show sensed values in real time. Students were also given a list of short questions which they needed to answer (mainly to verify that students had read the specification sheets of each sensor and understand how to set up each sensor). Extra credit was offered for exploring any of the other features available or combining sensor functionality.

5.4 Lab 3: Communication Part 1

During week 4 of lecture, Wi-Fi communication was introduced. Therefore, this lab was dedicated to test the various wireless capabilities of ESP 32 boards. Students created a wireless sensor network by programming the ESP 32 to act as a Wi-Fi access point and setting up a wireless mesh network of sensors. The network should have one node with a PIR sensor, a node with a BME680 sensor, and a node with a moisture sensor. The computer would listen to all the sensors in network and print the outputs to the website hosted on the node.

Students were instructed to set up a network with the following:

- The PIR node should only send a value if the PIR value changes.
- The BME680 and moisture sensor nodes should do the following: Sample every second, calculate the mean value for the last 60 s, and calculate the standard deviation for the last 60 s.
- The BME680 and soil moisture sensor samples calculate mean and standard deviation every second based on data for last 60 s. They send data only if new values differ by one standard deviation from the current mean.

To receive credit for the lab, students needed to successfully explain and demonstrate that the above wireless sensor network was working properly.

5.5 Lab 4: Communication Part 2

Lab 4 continued building on the low power and IoT for the home topics covered in class. This Lab was designed to give students practice with the LoRa capabilities of the provided ESP32. Students were asked to make a long range moisture sensor which also provided GPS coordinates. To receive credit for the lab, students needed to create a node that is able to measure and transmit soil moisture every 5 s to another node. The node should send the moisture reading along with a GPS location of the node. Students were also asked to complete simple questions to reinforce concepts about LoRA vs LoRaWAN communication, transmit capabilities, and how to improve energy efficiency.

5.6 Lab 5: Communication Part 3

Lab 5 was paired with the in-class discussions on low power wide area networks and IoT for the home. Students test the Bluetooth Low Energy capability the ESP32 modules. This lab gives an introduction how to create a simple BLE device and server.

Students were familiarized with GATT (Generic Attributes)—BLE Service and BLE Characteristic. Students learn how to work with them to build a BLE client and server network. Students used the BLE server and a BME680 node to send temperature value to a BLE client connected with a relay. Students needed to demonstrate a client which examines the temperature value of the server node and turns on a relay if the temperature goes above some threshold value.

Extra credit was offered for extending this project. One interesting idea is having the client connect to another commercially available device such as a heart rate monitoring watch.

5.7 Lab 6: Management

This lab was designed to reinforce the IoT management topics covered in lecture. In this lab, students learned how to interface with the MQTT broker and how to publish/ subscribe from an ESP32 node. Students also created and used Amazon AWS free IoT services.

The lab was focused to register the IoT device (BME temperature sensor) and going through the steps to use the free IoT AWS services like creating security credentials, adding policy, connecting with MQTT broker using ARN, adding the broker address to the AWS account. Finally, students needed to publish the sensor values at regular intervals and have the sensor subscribe to itself to determine how often to sample the temperature.

To receive credit for this lab, students needed to show their working sensor. Extra credit was available for having the ESP32 subscribe to a second service that is responsible for adjusting the sampling rate of the node.

5.8 Lab 7: Security

The LoRA library and other labs completed previously do not implement security; everything is transmitted openly. In this lab, students build upon the security topics discussed in lecture and learn how to use encryption to secure transmissions.

Students modified the code from the previous LoRA lab to utilize RSA encryption/decryption. Students learned how to secure the data transmitted in LoRa packets using symmetric, asymmetric, and LoRa encryption approaches. In this case, the transmitting node encrypted data and sent data to a receiver node, where the message gets decrypted and printed on the serial console.

To receive credit for the lab, students needed to demonstrate successful encryption, transmission and decoding of a message. Extra credit was offered to students who were able to generate an AES key on the fly and use RSA to pass the AES key to the other node/nodes.

5.9 Lab 8: Visualization

Lab 8 was designed to reinforce the data analytics topics covered during lecture. Students are asked to create an ESP32 program that outputs serial values of a sensor of the students' choosing. Then, students connect the ESP32 to a Raspberry Pi and create a Python script that pushes the data from the ESP32 into the MySQL database continuously. This data should then be displayed on a website using PHP to pull data from the database.

Students first set up the Raspberry Pi and install the LAMP(Linux, Apache, MySQL, PHP) stack. Then, students used Python scripts to read the serial data coming from ESP32 and push it to MySQL database. Finally, to receive credit for the lab, students needed to display the useful information collected in database on a webpage.

5.10 Additional Activities During Lab Sessions

In addition to the eight labs covered previously, some of the lab time was allocated for other tasks. One lab session in week 7 was used for elevator pitches, in which student groups shared their idea for the final project and received feedback (during each student presentation, the rest of the class was required to document and share anonymous feedback with the presenter). After the eight labs were completed, the rest of the lab slots were reserved for additional lectures. During the last week of the semester, the entire week was reserved for “open lab” to allow students extra office hours and time to meet to get help on their final projects.

6 First Class: Spring 2019

6.1 Student Feedback

Reviewing feedback from students during and after the semester revealed additional insights and areas where we could improve. The feedback received from students was overall very positive. The labs were overall very well received and students found the hands-on experience to be new, interesting and valuable. Many students commented that the hands-on experience in the labs was very helpful for reinforcing and learning concepts from lecture. Some students decided to use their IoT projects as a capstone or thesis project. One student was even able to secure an internship in IoT, in part due to having gained a basic understanding of IoT from taking the course.

Students did point out several areas where the course could be improved. Some students commented that it was sometimes challenging for group members to meet outside of class in order to complete the final project and to complete the more challenging lab assignments. Other students would have preferred to complete the projects individually rather than in groups, and to have more office hours to get help.

7 Second Class: Spring 2020

7.1 *New Updates and Changes Made*

Based on the feedback from the previous semester, several key changes were made to improve the course. These included the following:

- Updating grading structure: We kept the assignments and peer evaluations, but replaced the midterm exam with weekly online quizzes. This allows students more opportunity to influence their individual grade.
- Adding extra “open lab” time for final project: Removing the midterm and final exam timeslots frees up two more timeslots for students to work on their final projects with their groups.
- Implementing weekly online quizzes: Quizzes were posted online each week, designed to check for understanding on basic lecture material. Weekly online quizzes also incentivized students to learn course material and provided an opportunity for students to have more control over their own grade rather than relying heavily on group projects.
- Minor changes to labs for clarity: the labs were overall well received, so we kept the labs the same except for making changes to improve clarity.

7.2 *Lessons Learned*

One key item we learned from offering the class is that having a dedicated IoT space would be very helpful. Student groups needed to remember to pack and bring their component kits to each lab. Also, student groups would have benefited greatly from having more work space to meet during non-class hours to finish projects. While we were able to successfully run the course without having this additional space, we highly recommend making additional space available for students to store their kits and meet to work on projects. Since the student population at East Bay has many students who commute, this shared space would be extremely beneficial.

Even more ideal would be having a space where students could get help on their prototyping projects, and offering additional office hour time to support the students. In the course evaluations, several students commented that additional office hour time would have been beneficial. Often, a simple wiring error or small programming error would cause groups to be stuck for long periods of time. Students frequently used office hours to get help and found the sessions very useful.

We also found that students would likely have the best experience in the class if they are able to work either a group of two students or individually. Once the group of students became larger than two, it becomes more challenging for each student to gain as much hands-on practice and learning with the kits. Due to budget constraints, we were only able to have groups of two to three students. If smaller groups are

utilized, we would also recommend allocating extra time for project demos and office hours, since meeting with additional groups does require more time from the instructor and assistants.

8 Plans for Next Class

While our first attempts at teaching IoT were very well received, we will continue to refine and improve the course. One area we would like to explore is creating an IoT “part two” or capstone course. Once students have learned the fundamental skills in the introductory course, more advanced skills and projects could be covered in a future course. Having a year-long course could also provide students and instructors the opportunity to dive deeper into certain areas and applications of IoT, such as medical, security, networking and data analytics.

Based on feedback from students, there is great interest in learning more about hardware and IoT areas. With growing demand for IoT, we could explore adding an IoT concentration, certificate, or minor to our computer science curriculum. Providing unique projects and experiences through the IoT curriculum also enables students to gain new, unique experiences and projects. These opportunities can help students improve their resumes and career competitiveness, be able to complete unique capstone projects, and better prepare students for challenges faced in designing real-world IoT projects.

9 Conclusion

In this work, we have shared our process for designing, planning, and teaching an IoT course for advanced undergraduate and graduate level students. Based on our experiences, we believe that combining hardware and software approaches is an effective way to introduce computer science students to IoT concepts. In particular, we found that introducing material during lecture, followed by hands-on practice in lab provided students a richer learning experience. Many of our students had never been exposed to hardware prototyping and debugging, yet these students were able to successfully complete the assignments and projects and found the course to be a positive and valuable experience. One student reported being able to secure an internship in IoT, due in part to having completed the course and having a solid introductory background. Other students were also able to use their IoT projects toward capstone and thesis projects.

For both students and instructors alike, we have found the project-based, hands-on learning strategies in our course to be powerful and effective in helping students not only learn IoT concepts but also apply them to new projects. We anticipate that these strategies will be beneficial to many students and educators who are interested in the IoT field. One of the key challenges we encountered while developing our

course was identifying hardware and equipment to use in the course, and integrating it successfully into labs and assignments. By providing detailed lists of our hardware components, topics covered, labs, and course structure, we anticipate that we can help others continue to develop and improve IoT education.

Acknowledgments The authors gratefully acknowledge CSU East Bay’s A2E2 program for funding hardware kits for the IoT course.

References

1. That ‘internet of things’ thing (2020). <https://www.rfidjournal.com/articles/view?4986>. Accessed 05 April 2020
2. Worldwide semiannual internet of things spending guide. IDC (2019)
3. Worldwide internet of things forecast, 2019–2023. IDC (2019)
4. Internet of things: Foundations and applications (2020). <https://www.ischool.berkeley.edu/courses/info/290/iot>. Accessed 05 April 2020
5. Internet of things (2020). https://www.nwmissouri.edu/csis/msacs/PDF/Syllabus/44-599_IoT_Syllabus.pdf. Accessed 05 April 2020
6. J.W. Kang, Q. Yu, E. Golen, Teaching IOT (internet of things) analytics, in *Proceedings of the 18th Annual Conference on Information Technology Education*, ser. SIGITE ’17 (Association for Computing Machinery, New York, 2017)
7. Bus 5120 – securing the internet of things (2020). <https://msisuva.admin.virginia.edu/app/catalog/classsection/UVA01/1198/20125>. Accessed 05 April 2020
8. Interaction design studio II: Prototyping the internet of things (2020). https://catalog.csueastbay.edu/preview_course.php?catoid=19&coid=69309&print. Accessed 05 April 2020
9. Particle (2020). <https://www.particle.io/>. Accessed 12 April 2020
10. B. Siever, M.P. Rogers, Micro: bit magic: Engaging k-12, cs1/2, and non-majors with IOT & embedded, in *Proceedings of the 50th ACM Technical Symposium on Computer Science Education, SIGCSE 2019* (ACM, New York, 2019)
11. Arduino (2020). <https://www.arduino.cc/>. Accessed 12 April 2020
12. Arduino (2020). <https://www.raspberrypi.org/>. Accessed 12 April 2020
13. Particle photon (2020). <https://docs.particle.io/photon/>. Accessed 12 April 2020
14. Nodemcu (2020). <https://github.com/nodemcu/nodemcu-devkit>. Accessed 12 April 2020
15. Esp8266ex overview (2020). <https://www.espressif.com/en/products/hardware/esp8266ex/overview>. Accessed 12 April 2020
16. M5stack fire (2020). <https://m5stack.com/products/fire-iot-development-kit>. Accessed 05 April 2020
17. Gas sensor bme680 (2020). <https://www.bosch-sensortec.com/products/environmental-sensors/gas-sensors-bme680/>. Accessed 12 April 2020
18. Gas sensor bme680 (2020). <https://www.broadcom.com/products/optical-sensors/integrated-ambient-light-and-proximity-sensors/apds-9960>. Accessed 12 April 2020
19. Freertos (2020). <https://www.freertos.org/>. Accessed 12 April 2020
20. Esp32 developer framework (2020). <https://github.com/espressif/esp-idf>. Accessed 12 April 2020
21. A new generation ecosystem for embedded development (2020). <https://platformio.org>. Accessed 17 April 2020
22. Visual studio code (2020). <https://code.visualstudio.com/>. Accessed 05 April 2020
23. Getting started with ESP32 Bluetooth low energy (BLE) on arduino ide (2020). <https://randomnerdtutorials.com/esp32-bluetooth-low-energy-ble-arduino-ide/>. Accessed 17 April 2020

24. ESP32 BLE client – connecting to fitness band to trigger a bulb (2020). <https://circuitdigest.com/microcontroller-projects/esp32-ble-client-connecting-to-fitness-band-to-trigger-light>. Accessed 17 April 2020
25. Adafruit ultimate GPS (2020). <https://cdn-learn.adafruit.com/downloads/pdf/adafruit-ultimate-gps.pdf?timestamp=1550967629>. Accessed 17 April 2020
26. Welcome to the home of mqtt.fx (2020). <https://mqttfx.jensd.de>. Accessed 17 April 2020
27. Register a device (2020). <https://docs.aws.amazon.com/iot/latest/developerguide/register-device.html>. Accessed 17 April 2020
28. Setting up your raspberry pi (2020). <https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/5>. Accessed 17 April 2020
29. Build a lamp web server with WordPress (2020). <https://projects.raspberrypi.org/en/projects/lamp-web-server-with-wordpress>. Accessed 17 April 2020
30. Welcome to pySerial’s documentation (2020). <https://pythonhosted.org/pyserial/>. Accessed 17 April 2020

Computational Thinking and Flipped Classroom Model for Upper-Division Computer Science Majors



Antonio-Angel L. Medel, Anthony C. Bianchi, and Alberto C. Cruz

1 Introduction

Graduation rates are generally low among STEM majors. According to an NSF study from 2003 to 2009, physical, computer, and math sciences have a retention rate of only 43% [1]. The California State University, Bakersfield (CSUB) is no exception. Four and six-year graduation rates are 16.2% and 40.9%, respectively, for first-time freshmen [2]. Increasingly disenfranchised students are dropping out of the program. A breadth of literature suggests that flipped classroom model improves student engagement in the classroom [3, 4], which in turn increases student performance and ultimately improves graduation rates. The impact of the flipped classroom model with respect to a control population is the focus of this study. This study aims at increasing student performance with peer-based, engaging teaching methods and computational thinking concepts.

A consideration is given to the two themes of computational thinking and flipped classroom model [4]. Computational thinking (CT) has revolutionized education [5], and it focuses heavily on teaching by example with peers. CT provides innovative teaching strategies for fields that are not even STEM, such as History and English. Paradoxically, in the field of CS, the medium of instruction is not CT. Traditionally, in upper-division core CS, the instructor states an algorithm, and class time is spent working backward to demonstrate it works. This is not constructive because the

Human subject testing protocol: IRB-exemption authorized (CSUB IRB 20–32).

A.-A. L. Medel (✉) · A. C. Bianchi · A. C. Cruz
Department of Computer & Electrical Engineering & Computer Science, California State University Bakersfield, Bakersfield, CA, USA
e-mail: amedel2@csub.edu; abianchi1@csub.edu; acruz37@csub.edu

instructor gives the solution to the problem and does not invite independent thought about how to solve it. Though the topic is computer-related, students do not engage using CT concepts. Building on a CT framework, our flipped classroom participants complete peer-based in-class activities that work forward, not backward. They must develop a good approach (often pseudo-code) to solve the problem on their own without being presented a solution. The main goal of this work is to develop peer instruction/pair programming [6] lessons for computer science classes—with intent to open source the curriculum at the conclusion of the study—and to assess and evaluate the impact of the lesson plans. It is anticipated that the flipped classroom model will improve student performance and ultimately increase graduation rates.

2 Background and Related Work

CT and active learning led to a boom in pedagogical methods. However, most works are in fields other than CS: K-12 and non-major CS lower-division coursework (such as CS0). A recent survey [7] highlights 27 works focusing specifically on K-12 and some higher education on integrating CS concepts in the classroom. There were 16 quantitative studies in this survey and all report results in favor of the experimental groups using CT. In another study, a 2-year project in Italy disseminated breaking technologies in K-12, injecting computational activities in the classroom [8].

Flipped classroom is a method where traditional lecture is replaced by collaborative problem-solving activities [9]. Time spent outside of class, where an individual normally completes homework, is replaced by self-paced videos and interactive lessons. This is the opposite of the traditional approach, hence the name *flipped classroom*. A recent review of 32 studies [3] found that the flipped classroom model has the potential to increase learning performance, improve attitude, enable more discussion, enforce cooperative learning, and improve learning habits. Yet, the impact of social/peer instruction is understudied among *upper-division* CS major classes. This study addresses the following research questions:

1. How does a flipped classroom compare to a control class that is already highly online, accessible, and engaging (education management system, Kahoot! activities, online homework, and the availability of prerecorded lectures)? How much of the benefit of a flipped classroom model is due to the use of advanced digital pedagogical techniques/systems?
2. How does instructor competence in CT, active learning, and flipped classroom affect the experience in the classroom?
3. Does the flipped classroom model improve frequency and quality of social interaction between the student and other students and the instructor(s)?
4. Does CT, active learning, and flipped classroom positively impact academic performance in upper-division CS when compared to a control group?

2.1 Contribution to State of the Art

This study is framed as an ablative study with data collected from a control/non-flipped population in fall of 2019 and from a flipped classroom population in the spring of 2020. The environment is highly controlled (same instructor, syllabus, pace, and curriculum). This is novel since most other works do not provide a comparison to a control population and often do not focus on upper-division CS courses. To the authors' knowledge, this work is the first to provide a case study for undergraduate Artificial Intelligence.

3 Approach and Methods

Students are split into two groups: the control (CON) and the flipped classroom (FC). CON receives traditional instruction (upper-division core Computer Architecture II and Artificial Intelligence, with 39 students in total). FC receives Just-In-Time-Teaching (JiTT) [10] and pair-programming activities [6] (same courses, 40 students involved in total). Both groups use the same education management system (Moodle) and have access to past/prerecorded lectures and lecture notes. The CON group completes online homework after instruction, and the FC group completes online quizzes before meeting in class. Lab instruction is unaltered; the instructor gives a lab briefing, and students complete the lab with a partner of their choice. The instructor has completed the Berkley FLP program for STEM faculty and is competent in the areas of CT, active learning, and flipped classroom.

3.1 Control (CON) Population

CON students complete weekly homework assignments on material covered in lecture. Homework consists of 20–25 MC and 3–5 free response (random bank). Lectures are students' first exposure to the material. The instructor provides an algorithm, and time is spent working backward to demonstrate correctness. Instruction is authoritative/non-dialogic [11], though students can ask questions at the conclusion of class. Day-to-day topics are fixed.

3.2 Flipped Classroom (FC) Population

FC students watch prerecorded lectures from the CON group and YouTube at their own pace with a short online assessment/quiz before each class. We use Just-In-Time-Teaching (JiTT) for assessment [10]. Quizzes ensure students are

participating. The depth and length of quizzes are less than CON group's online homework: 10–20 MC and 0–3 free response (random bank). The day's topics vary based on difficulties revealed by the JiTT test. Courses that implement JiTT have had a myriad of variations, yet the results have been small to significant improvements [10].

There is no traditional homework. In lecture, there is a short authoritative/dialogic [11] lecture of less than 15 minutes—sometimes none if the lecture is a continuation of previous topic. Students then complete active learning/CT activities to use their working memory to interrupt the forgetting curve and commit the new knowledge into long-term memory [12]. In our study, we use pair-based worksheet activities. The worksheets are collected and graded for correctness. They replace lecture in a traditional classroom and daily clicker quizzes [13]. Students are assigned to a specific partner and complete the worksheet with pair programming [6]. In *pair programming*, one student is the driver and the other, the navigator. The driver focuses on the problem at hand and is the only one writing/coding. The navigator reads ahead, manages time, and supervises the work. As a tertiary benefit, students learn pair programming which is often used in the industry. The worksheets are a long arc where students construct a solution/algorithm, rather than work backward from an existing solution (see Fig. 1). The pair proceeds at their own pace.

<p>Names (Driver/Navigator):</p> <p>IDs:</p> <p>Date:</p> <p>Unscored Problem 1 (Discussion): Consider the following: 0x409C add \$a0 \$a2 \$t0. Discuss with Navigator the type of operation, convert it to binary, and what components are used by this operation.</p> <p>Problem 2 (Free response): Sketch the PC logic. What are the inputs? What is the result?</p> <p>Problem 3 (Free response): Register file contains: \$0 : 0x0000 ... \$31 : 0x0013 Sketch the register file. What are the inputs? What are the outputs? ...</p> <p>Problem 7 (Free response): Connect the dots. Draw a complete circuit of the processor as it completes the operation.</p> <p>Problem 8: Repeat this worksheet for the following operation: srl \$v0, \$t0, 3</p>

Fig. 1 Pair-programming worksheet for Computer Architecture II. Students brainstorm the data path as it executes a MIPS R-type instruction. Conventionally, students are presented with a circuit of the processor. With this scaffolded activity, students build the circuit from the ground up

The pairs are random each day, with the class stratified into three equally sized tiers based on class performance (low, mid, and high students). Students are paired with individuals of their own tier, to avoid towers of knowledge [14] and ensure that each pair is working in their zone of proximal development [15].

3.3 Data Collection Reported in This Work

Surveys are administered at the beginning and end of the semester to gauge student social interaction with other students, the instructor and instructional student assistant, and their opinions about the class. This normalizes existing student biases and opinions of the instructor gained over the semester. Students are also asked if they have repeated this class (between fall and spring) and if they have participated in a class involving active learning, CT, and/or flipped classroom model. Surveys consist of 7-point Likert items with additional free-response questions for students to further elaborate to avoid issues such as acquiescence bias, often encountered with Likert item questioning. See Table 1 for the number of participants who gave informed consent. The number of participants in the study (the total number of students in each class) is less than the number of students who gave informed consent for data collection.

3.4 Data Collection Not Reported in This Work

We also aggregate midterm and final exam performance for both CON and FC groups, though this data is not included in our preliminary results. The exams are the same for each group. This measures students’ academic performance and compares the two groups to identify if there is a gain in academic performance in the experimental group. Homework and JiTT quizzes are too dissimilar to compare and are based on a random bank of questions.

We also administered Kahoot! quizzes throughout the semester based on the protocol established in [13]. Students respond to a battery of questions as a group and then are asked to repeat the same questions on an individual basis to normalize group biases. Performance in this activity can gauge individual vs. group performance and measure quality of social interaction. Kahoot! data collection was suspended due to COVID-19 considerations and will be reported in later work.

Table 1 Total number of participants that gave informed consent from each group

	Control (CON)		Flipped classroom (FC)	
	Pre-survey	Post-survey	Pre-survey	Post-survey
#	22	17	22	12

The number of participants in the surveys is less than the total number of students in the class

4 Preliminary Results

Our preliminary findings include student comments from the CON group and FC group and some implementation thoughts for those considering a flipped classroom model. During data collection of the FC group, the university converted to alternative delivery in response to COVID-19. *Alternative delivery* is the phrase the California State University (CSU) uses referring to the transition from in-person classes to synchronous online classes. Virtual instruction was effective on March 17, 2020, 9 weeks after the start of the semester. For reference, a complete semester is 16 weeks. Post-surveys for the FC groups were likely affected by the transition to alternative delivery. With alternative delivery, the class was converted to a synchronous online class using traditional pedagogical teaching methods, though students were instructed that the study should focus on the time spent prior to alternative delivery with flipped classroom model. Any results presented in our work should be considered preliminary for the following reasons:

- (a) FC group did not participate in flipped classroom model for an entire semester.
- (b) Though FC group was instructed to provide comments on the flipped classroom version of the class (before alternative delivery), it is possible that some respondents disregarded this instruction.
- (c) We consider the number of participants in our study to be too low to establish significance.

Thus, we plan to continue the study for the next academic year to present strong evidence of our conclusions in later work.

4.1 Considerations for Instructors

Some of the students claimed to have participated in a flipped classroom model at CSUB or at a local community college, yet either an instructor did not provide online videos for viewing—basing activities from reading alone which is not enough in our opinion, or did not implement JiTT style quizzes, which are a critical feedback mechanism to ensure student success. Considering this dissimilarity, students informally stated that they preferred the model presented in our study to previous encounters with the flipped classroom model.

Further, at-risk students in FC appreciated the flipped model, had higher attendance rates, and did better on worksheets. Students repeating the class often did not complete homework leading to their failure in a prior semester though when participating in the FC group, their attendance and assignment completion improved. However, high-achieving students did not view FC as positively as at-risk students. Some students tried to complete worksheets on their own in a rejection of the pair programming structure and/or complete the worksheets as quickly as possible with little social interaction. Often, these groups would segment the day's

work into discrete parts and solve them separately. Generally speaking, flipped classroom model seemed polarizing, and students either strongly liked it or strongly disliked it (note that this is anecdotal, based on instructor and teaching assistant observations; cessation of in-class activities required us to suspend data collection, so we plan to obtain more data to support these claims at a later point).

4.2 *Qualitative Preliminary Results from CON Group*

Most students in the CON group indicate that they study alone for classes:

1. Reading the textbook and reviewing lecture material, usually alone.
2. I prefer to study and prepare for a class in solitude . . .
3. [sic] [prefer to study] by myself, with the help of the professor~~

In both pre- and post-surveys, students prefer studying alone, and most likely do not automatically form study groups. It is our hope that social activities in the classroom will continue outside of class. Learning is a social process [16], so an effective and engaged student population should have less individuals electing to study by themselves.

The CON group used some advanced online/digital pedagogical techniques that are often shared with a flipped classroom. To measure the quality of lecture, students responded to the question, “In the space provided, you may provide additional comments on the materials that are often given to you by instructor(s). Can they be improved? Were they effective?”, as a free response. Pre-survey responses indicated that no professors at CSUB provide taped lectures, with one student using YouTube to seek supplementary video material. Post-survey responses often praised the availability of taped lectures:

1. Instructors’ videos are amazing; they care enough to put in extra effort to ensure we are never lost.
2. Would be nice if [other] professors would program examples on projector and record them more often.
3. Most professors should do video recordings because it helps. Having a detailed syllabus of what the class will be discussing in lecture daily helps.
4. Lecture recordings were very helpful.
5. This class is the perfect example of how to use current technology. All lecture notes available at the end of class, current updated grades, all resources in one place, easy communication with teacher, no excuse for not knowing the current status of the class.
6. [sic] *Lectures are recorded and edited * Notes always available [checked box] this class is quite good.

The CON group had access to recordings of previous lectures and responded well to this resource. In future work, we are eager to see how much of an improvement to the FC population was also due to taped lectures.

4.3 *Qualitative Preliminary Results from the FC Group*

Though the FC group had more than half of a semester to acclimate to a new teaching style and potentially experience the benefits of a flipped classroom, alternative delivery caused an abrupt end to our study. Students were asked to reflect on in-person activities prior to alternative delivery. However, their responses may not be totally void of some bias from dealing with the pandemic. They may have used the survey as an opportunity to vent frustration at synchronous online teaching. From comments left to free-response questions on the survey, the FC group enjoyed the time they spent in class. Having a more engaging environment with peers and the instructor gave students the confidence and reassurance that the professor is competent in the subject as well as confidence in their own work. A sample of free-response comments follows:

1. Not much [comments] in general. I lost motivation in my CS degree because of hostile professors or professors that just read off of slides.
2. Some professors provide great materials for their classes, and others just read PowerPoints written by other professors and do nothing else.
3. Interactions in other classes is close to none.
4. These comments showcase the issues with the traditional classroom.

Most students took the opportunity to contrast their FC experience with other control classes. There is a trend among students to not appreciate instruction that relies heavily on prepared materials (such as slides) and traditional environments that do not allow the students to speak to each other. There were two comments speaking positively about the flipped classroom model:

1. Before quarantine, I liked that [the instructor] would walk around during labs and activities to check our individual work.
2. With regard to [the instructors] flipped classroom. It was really good. Lectures can be much more engaging. Labs can incorporate subjects talked about in class more. I wish lectures didn't involve just vocabulary review.

With our flipped classroom model, the instructor would walk around the classroom observing student work and help as needed. Anecdotally, the ISA (instructional student assistant) for the Computer Architecture II section found that students were enjoying the interaction with the material and the instructor.

4.4 *Quantitative Comparison of CON and FC Using Likert-Type Responses*

Table 2 provides a summary of responses to Likert item questions for Artificial Intelligence and Computer Architecture II. For survey responses, students were asked to respond with strongly disagree (1), disagree, slightly disagree, neutral,

slightly agree, agree, and strongly agree (7) to a statement. The full survey included 33 questions, but we are hesitant to provide complete results because we feel that more survey participants are needed to support our conclusions. The surveys are Likert items and not a Likert scale. We use mode and frequency to aggregate responses to the Likert items [17]. Note that in general, the population size for post-surveys of CON and FC is so small that the answer distribution/frequency is sparse, and it is inappropriate to use statistical testing to determine significance (Chi-square tests). We hope to address this in later work with more data collection.

Research Question 1: Online/Advanced Traditional Classroom

We initially hypothesized that a significant part of the flipped classroom model was the availability of online resources and referring the students to taped lectures from previous classes. We ablated this aspect of the class by providing the CON group with access to the previous semester’s lectures.

In Table 2 Prompt 1c, when responding to tendencies to discuss the class with other students using the Internet, the CON group initially reported neutral, and at the conclusion of the class, this response shifted to strongly agree (with a frequency of 24%). The FC group initially responded with strongly agree, and the result changed

Table 2 Analysis of a subset of Likert-type questions from the survey on a 7-point scale, with 1 being strongly disagree/negative and 7 being strongly agree/positive

RQ	Q	Prompt	CON group		FC group	
			Pre	Post	Pre	Post
1	1c	I discuss the class with other students on the internet	4 (0.27)	6 (0.24)	6 (0.32)	5 (0.38)
1	1g	I review the notes or other provided material (videos, etc.)	4 (0.27)	5 (0.41)	5 (0.36)	4 (0.50)
3	1a	I prepare for class with a study group	3 (0.33)	2 (0.47)	1 (0.29)	2 (0.50)
3	1b	I discuss the class with other students	4 (0.36)	5 (0.41)	5 (0.27)	3 (0.25)
3	1d	I discuss the class with others who are not students (Reddit, etc.)	1 (0.50)	1 (0.82)	1 (0.50)	1 (0.50)
3	1e	I study for the class by myself	6 (0.36)	5 (0.24)	6 (0.41)	6 (0.50)
3	1i	I participate interactive activities during lecture	4 (0.41)	4 (0.47)	5 (0.30)	6 (0.63)
4	1f	I read the textbook	3 (0.32)	4 (0.59)	5 (0.27)	4 (0.38)
4	1h	I seek out material beyond what is provided by the instructor	4 (0.32)	5 (0.31)	5 (0.32)	5 (0.29)

Data aggregates responses from Artificial Intelligence and Computer Architecture II. FC group was affected by alternative delivery, and more data is needed to conclude the study

RQ most relevant research question, Q the question number, CON control group, FC flipped classroom group, Pre median response from pre-class survey, Post median response from post-class survey

Parenthesis indicates the frequency of the response

to agreement in the post-survey (with a frequency of 38%). We do not feel that either population has a frequent enough response to be significant.

In Table 2 Prompt 1 g, when responding to tendencies using review material provided by the instructor, such as videos, the CON group initially reported neutral, shifting the response to slightly agree in the post-survey (with a frequency of 41%). Antithetically, the FC group response to this question changed from slightly agree to neutral. Students were expected to watch online videos to complete JiTT quizzes and prepare themselves for the class. More data and more specific survey questions are needed in the future to determine what resources the students turn to if they are not using the online videos to complete the JiTT quizzes.

Research Question 2: Perception of Instructor Confidence

Perceptions of instructor confidence are relevant to any instructor considering the adaptation of a flipped classroom model. The instructor involved in this study received training in computational thinking, active learning, and flipped classroom model from the Transforming STEM Teaching Faculty Learning Program offered by UC Berkeley's Center for Teaching & Learning and should be considered an expert. However, this may not be the case for most instructors. Flipped classroom model has not been widely adopted in CS, with only 7.6% of all students indicating prior experience with this classroom model. Potential instructors may be as much of a novice as the students themselves when it comes to flipped classroom model. As such, there would be concerns of highly negative perception of the instructor by the FC group, despite their expertise. Negative perceptions would lead to negative classroom observations and instructor ratings, potentially impacting retention of the instructor and discouraging implementation of the flipped classroom model. There was insufficient survey participation for questions relevant to Research Question 2, and we hope to conclude this in later work.

Research Question 3: Improved Social Interaction

We anticipated an increase in social interaction for students with a flipped classroom model. To support this claim, we would expect to see a decrease in the number of students indicating that they study alone and the difference to be greater in the FC group.

Table 2 Prompt 1a confirms our hypothesis. With the CON group, students slightly disagree with the statement that they form study groups, changing to disagreement at the end of the semester. Though the FC group was initially in strong disagreement that they form study groups, there was a shift in responses to disagreement (with a rate of 50%). We believe this to be a significant result. Flipped classroom model may slightly improve student attitudes to study groups, though in general students do not engage in this behavior.

In Table 2 Prompt 1b, 1e, and 1f, results contradict our hypothesis. The CON group saw an increase of students discussing the class with other students, and the FC group saw a decrease. Though, the frequency of the response casts doubt on its significance. The CON group had fewer students who indicated that they study alone, and the FC group saw no change (with a high frequency of 50%). The CON group had a significant number of students who indicated that they read the textbook (59% frequency), whereas the FC group did not give a strong/significant response. In Table 1 Prompt 1d, there was a consensus between CON and FC groups that they did not seek help from non-students during the class with a very high frequency.

Overall, FC responses to these questions may have been adversely affected by students who were not following instructions to reflect strictly on the portion of the class before alternative delivery. It should also be noted that there was a low response rate for post-surveys (see Table 1). In general, more data is needed, and we intend to continue our study in the next academic year.

Research Question 4: Student Performance

Student performance can be best measured by classroom performance which will be reported in later work. Performance can also be measured by the rate at which students engaged in positive study behaviors, such as reading the textbook (Prompt 1f), and their tendency to seek material beyond what is normally provided by the instructor—indicating piqued general interest in the class topics (Prompt 1h).

Considering Table 2 Prompt 1f, the CON slightly disagreed with the concept of reading the textbook and later indicated that they are neutral about it in the post-survey (with a rate of 59%). In contrast, the FC group had slight agreement that downgraded to neutral in the post-survey. It is possible that FC students preferred to watch the lecture videos rather than read the textbook.

Considering Table 2 Prompt 1h, students indicated the rate at which they seek out material beyond what is provided by the instructor. Both the CON and FC groups indicated they were somewhat likely to engage in this behavior, but there is not a significant difference between the groups.

5 Conclusion

With a flipped classroom model, students use their working memory to apply old knowledge to new, more complex problems with their peers. This creates a deeper understanding of the material, and students retain the knowledge longer by interrupting the forgetting curve. Our study is ongoing, and we have collected data for the control group and a little more than half a semester of the experimental group due to transition to alternative delivery. Nonetheless, from our preliminary results, we have found some insights: flipped classroom model is viewed more positively by at-risk students, viewed less positively by high achievers, and that

students in the control population already report better engagement with access to online prerecorded lecture videos. Our anecdotal insights and survey free-response comments were contradicted by the quantitative results in our Likert item analysis. We are still confident that our hypothesis will meet expectations. Quantitative analysis of our data at this point suffers from a low number of samples and sparse distribution of responses that will be addressed in the future with more data collection when the university resumes conventional operation. We hope that future results will provide a coherent conclusion about the effectiveness of the flipped classroom model.

References

1. National Science Foundation, Science and Engineering Indicators Report (2014)
2. The offices of Institutional Research, Planning, and Assessment; Public Affairs and Communications, 2018. [Online]. Available: https://www.csub.edu/irpa/_files/FACT-BOOK-FINAL.pdf. Accessed 11 Apr 2019
3. M.N. Giannakos, J. Krogstie, Reviewing the flipped classroom research: Reflections for computer science education, in *CSERC '14: Proceedings of the Computer Science Education Research Conference*, (2014)
4. A. Roehl, S.L. Reddy, G.J. Shannon, The flipped classroom: An opportunity to engage millennial students through active learning strategies. *J. Fam. Consum Sci* **105**(2), 44–49 (2013)
5. J.M. Wing, *Computational Thinking* (Microsoft Research Asia Faculty Summit, Tianjin, 2012)
6. N. Nagappan, L. Williams, M. Ferzli, E. Wiebe, K. Yang, C. Miller, S. Balik, Improving the CS1 experience with pair programming. *ACM SIGCSE Bulletin* **35**(1), 359–362 (2003)
7. S.Y. Lye, J.H.L. Koh, Review on teaching and learning of computational thinking through programming: What is next for K-12? *Comp. Hum. Behav.* **41**, 51–61 (2014)
8. I. Corradini, L. Michael, E. Nardelli, Computational thinking in Italian schools: Quantitative data and teachers' sentiment analysis after two years of Programma il Futuro, in *2017 ACM Conference on Innovation and Technology in Computer Science Education*, p. 2017
9. B. Tucker, The Flipped Classroom Online instruction at home frees class time for learning, 2012. [Online]. Available: https://www.msuedtechsandbox.com/MAETELy2-2015/wp-content/uploads/2015/07/the_flipped_classroom_article_2.pdf. Accessed 21 Apr 2020
10. G.M. Novak, Just-in-time teaching. *New Dir. Teach. Learn.* **128**, 63–73 (2011)
11. S. Lehesvuori, J. Viiri, H. Rasku-Puttonen, Introducing dialogic teaching to science student teachers. *J. Sci. Teach. Educ.* **22**(8), 705–727 (2011)
12. The Human Memory, 2019. [Online]. Available: <https://human-memory.net/short-term-working-memory/>. Accessed 20 Feb 2020
13. C.B. Lee, S. Garcia, L. Porter, Can peer instruction be effective in upper-division computer science courses? *ACM Trans Comp Educ (TOCE)* **13**(3) (2013)
14. K.M. Lui, Pair programming productivity: Novice–novice vs. expert–expert. *Int. J. Hum-Comp. Stud.* **64**(9), 915–925 (2006)
15. J. Wertsch, The zone of proximal development: Some conceptual issues. *New Dir. Child Adolesc. Dev.* **23**, 7–18 (1984)
16. J. Grusec, Social learning theory and developmental psychology: The legacies of Robert Sears and Albert Bandura. *Dev. Psychol.* **28**(5), 776–786 (1992)

A Dynamic Teaching Learning Methodology Enabling Fresh Graduates Starting Career at Mid-level



Abubokor Hanip and Mohammad Shahadat Hossain

1 Introduction

According to Forbes Magazine, many of the skills that employers now require “are technical, involving proficiency in industry-standard or job function-standard software,” with “technical skills now outnumber[ing] all other competencies (cognitive and non-cognitive) in job descriptions across virtually every sector of the [U.S.] economy” [1]. Accordingly, one might expect that most employers are investing extensively in in-house training programs for entry-level IT graduates. In reality, however, nearly the opposite is true. In fact, “American employers have never been less interested in training new hires” [2]. Based on such reports, we must conclude that new IT graduates are facing greater difficulties today than ever before in acquiring entry-level IT positions. Employers typically now require new IT candidates to already have at least 2 years of professional IT experience. Thus, in terms of actual skill sets, there is a great need in the IT sector to bridge this wide gulf between what school and university IT programs teach and what business, industry, and government employers actually require.

Although we are now living in the age of the Fourth Industrial Revolution (“4IR”), universities today are still typically run by a 500-year-old model that overwhelmingly stresses theory over practical mastery of actual job tasks. In other words, the educational system has yet to develop the appropriate teaching and

A. Hanip

PeopleNTech Institute of Information Technology, Inc., Vienna, VA, USA

e-mail: ahanip@peoplentech.com

M. S. Hossain (✉)

Department of Computer Science and Engineering, University of Chittagong, Chittagong, Bangladesh

e-mail: hossain_ms@cu.ac.bd

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_16

229

learning pedagogy to cope with the practical job demands of the 4IR workplace. This skills deficit can be overcome, however, by creating an innovative “ecosystem” that seamlessly integrates the educational and entrepreneurial settings in today’s IT economy. Accordingly, this new smooth integration of both worlds enables academe and industry to work as a team, instead of at loggerheads. Unfortunately, at present, we are really lacking this practical approach in today’s university pedagogy. This is the chief reason that our IT job candidates remain unemployed for so long after graduation. Universities have an important role to play to ensure both optimal use of academic resources and prompt, reliable delivery of actual employment value in exchange for students’ hard-earned (or borrowed) tuition money.

For recent graduates to obtain mid-level positions right out of university or technical school, they must demonstrate expertise in all of the required skill sets, not just mastery of theoretical coursework. They must also understand and show professional behavior on par with industry expectations for experienced professionals. Candidates for such positions must be prepared to deliver right away without needing extensive training by industry. Moreover, during the actual employment interview, IT applicants must actually be able to prove their competence in the required practical job tasks. Finally, they must also demonstrate knowledge of corporate etiquette in order to be hired. Such testing is part of the hiring process.

Therefore, the practical part of university training necessitates addressing all the “three domains of learning”: cognitive (knowledge), psychomotor (skills), and affective (attitudes) [3]. The most important consideration in implementing this approach is determining how and to what extent these three aforementioned domains of learning must be emphasized in our various types of “pre-employment curricula.” The end result can achieve expectations only if the objective can be identified from current job requirements-based need analysis. Skills competency assessment regimens can be devised using appropriate methodology and precisely targeted resources. Quantifiable outcomes must be evaluated with reference to preset concrete performance standards.

The following employment-oriented curriculum design features should be used for a typical skill development training program.

- **Precisely assess students’ educational/functional skills deficits:** This can be done through needs analysis, which involves the collection and analysis of data related to the learner. Importantly, the data related to the requirement for the job which has been in target of the training is essential. This can be obtained from the hiring requirements of skill and expertise of a certain years experienced person. The data might include what learners already know and what they need to know to be proficient in this particular area or skill. It may also include information about learner perceptions, strengths, and weaknesses.
- **Keep adult learning styles in mind:** Keeping adult learning styles in mind is an important enough topic to address here on its own. Adult learners share certain characteristics that make training more effective for them. Curricula must recognize and respect the fact that these adults want to learn job-oriented knowledge and skills and that they tend to be self-directed. Such students often

bring to the classroom years of prior knowledge and job experience in other fields when they enroll in courses to enhance their practical skills in the IT industry. Such students also tend to be goal-driven and thus want their new training to be relevant and task-oriented. These students learn best when they are motivated to learn, and nothing stimulates such motivation more than actual classroom delivery of useful skills that a student knows will make him an asset to any company. Accordingly, IT training is likely to be more effective when these utilitarian principles are considered when designing curricula.

- **Create a clear list of learning goals and outcomes:** Naturally, companies hiring IT professionals have their own lists of expectations for new hires—both functionally (job–task competency) and behaviorally (professional etiquette). Since companies routinely complain that there is typically a large gap between what they expect of applicants and what applicants actually bring to the table, one cannot overstress the need for educational institutions to “do their homework” in this regard when designing courses and conducting classroom instruction and lab practicums. Theory is nice, but practical knowledge is increasingly what both business and government employers are seeking. Hence, those institutions that can crank out the highest percentages of “competency-vetted” graduates will rise the highest in the rankings that matter most to the corporate world. A key part of the equation in this regard is not only that educational institutions cater to practical skills requirements of industry, but also that the course curricula stay absolutely up to date in terms of the latest, “cutting-edge” changes and trends in technology.
- **Identify constraints:** Quite apart from the aforementioned considerations relevant to optimizing curriculum design, institutions should also be sensitive to constraints on resources. For example, a student’s time is itself a scarce and limited resource; hence, time scarcity should affect the design of degree programs and their course design. After all, students have only so many hours, days, weeks, or months in an academic quarter, semester, or year, and only so many years in a degree program. Another type of constraint that must be considered in curriculum design is the practical fact that students have only limited financial resources available to devote to the pre-employment educational phase of their lives. In other words, “bang for the buck” is an issue. Students increasingly calculate the return on their investment in terms of employment and salary in relation to cost of tuition. Consequently, there is a true market need for pre-employment education that bridges the gap between classical education and practical preparation needed in order to “win” in the great game of “musical chairs” that is today’s job application process.
- **Utilitarianism is king—continually identify and optimize instructional methods and materials:** It is essential to view the quality of an institution’s educational offerings from a utilitarian standpoint. That is, administrative and instructional decision-makers should continually analyze and evaluate their institution’s and their instructors’ “performance” in terms of how readily their students are meeting their own career goals after graduation. In a practical sense, this means, for instance, that if certain pedagogical methods and/or

materials prove not optimally conducive to enhancing student mastery of practical knowledge and skills, then the design of the corresponding instructional materials, teaching style, and/or curriculum structure must be altered accordingly.

- **Establish concrete performance evaluation standards:** Great care must be given to how student achievement is evaluated again with a utilitarian eye toward practical efficacy. The two most important evaluative criteria are student competency/performance and student behavioral aptitude in relation to employers' expectations. Functional competency and social skills will, after all, determine the degree to which the graduates succeed in securing at least mid-level employment soon after graduation. Accordingly, the most effective form of evaluating student performance is ongoing and summative [4].

Owing to the absence of sufficiently effective cooperation between educational institutions (both academic and technical) and employers, students are generally unable to obtain enough practical IT knowledge before graduation. Internships tend to focus on entry-level menial tasks and thus do not suffice as an adequate practical adjunct to traditional university—or even technical school—classroom instruction. Yet such practical real-world experience and competency are increasingly crucial for obtaining a first job in today's market. Moreover, especially with university IT instruction, the course curricula are not updated often enough or carefully enough to train students in a way that delivers the skill and knowledge levels that most employers require. Consequently, such under-trained graduates do not get the opportunity to obtain proficiency in the latest software tools and technology that business, government, and industry are using. But it does not have to be this way! Indeed, what if the educational curriculum-makers decide to require students to undergo rigorous lab-type training that accurately and authentically replicates the outside work environment? [5].

2 How One IT Training Institute Pioneered a Way to Fill the Skills Deficit and Place 95% of Its Graduates in Mid-level IT Jobs Within 4 Months of Graduation

A U.S.-based IT professional skills development institute called PeopleNTech has innovated a 4-month program whose graduates routinely land mid-level or senior-level IT employment shortly after graduation. The first step in accomplishing this goal is to narrow a student's focus within the industry. In other words, instead of trying to turn out a student who is a "jack of all (IT) trades, but master of none", PeopleNTech's approach steers students to first decide which sub-specialty (networking, programming, database, etc.) in which they wish to acquire deep, specialized knowledge. The curriculum then requires practical lab instruction in the desired specialization (see Fig. 1).

By giving this key, sevenfold key formula for employment success to all graduates of our institute—as the chief focus of their curriculum—PeopleNTech has

Fig. 1 Learning components identified by PeopleNTech for mid-level job readiness



been successful in placing over 95% of our graduates in mid-level and senior-level positions in the IT industry over the last 15 years.

Accordingly, given PeopleNTech’s extraordinary industry track record, this chapter focuses on explaining our teaching and learning methodology. Next, the discussion is followed by sharing our use of what is known in the real world as a “Belief Rule-Based Expert System” (BRBES), which actually assesses the overall skill level of a student by conducting aptitude tests according to the criteria mentioned in Fig. 1. Traditional skills assessments cannot adequately quantify a student’s performance attainments and overall skill levels, yet the BRBES approach can do so, which is why PeopleNTech uses this specialized assessment system. The institute quantifies a student’s skill and knowledge levels both upon enrollment and at graduation.

PeopleNTech is able to utilize the best award-winning instructors throughout its program in order to provide the highest quality of training, while also keeping costs down for both individual students and corporate clients. PeopleNTech’s curriculum and teaching are managed by industry professionals backed by years of corporate IT experience. PeopleNTech designs, implements, and manages workforce and partner-development programs for individual students and all sizes of companies up to Fortune 500 firms. PeopleNTech’s utmost priority is to keep its staff, students, curriculum, and facilities up to date with the latest innovations in technology, while embracing the latest industry trends.

2.1 Researching Latest Industry Trends

Continuous research on the trends and requirements of the current market is essential. This ongoing research is facilitated by more than 6000 PeopleNTech alumni working in the IT field. All PeopleNTech instructors have professional IT industry experience, and senior members of the hiring team, the recruiters who are working across hiring world of USA, our own marketing team, and job placement team are the source of data for need analysis.

2.2 Classroom Template

Providing quality practical, targeted instruction in a classroom setting is the foremost requirement for creating employable IT graduates. PeopleNTech creates an accurate replica of a typical industry work environment for each of its courses. This “lab approach” creates the quickest “fast track” for students to advance their career or to start a new one.

The company is equipped with certified instructors, who are active practitioners and true masters in their fields. In addition to teaching technical know-how, they also impart their knowledge of corporate culture and etiquette. Extensive hands-on lab exercises and state-of-the-art training facilities create a powerful learning environment that optimizes every student’s professional success beyond graduation. PeopleNTech also takes care to select industry-leading course manuals and reference materials. These resources help to facilitate quick acquisition of the substantive knowledge required by the industries, which is not possible through the traditional university education system—simply because its faculty members are less exposed to real-world conditions and expectations.

2.3 Essential Instructional Elements

PeopleNTech’s innovative pedagogical formula includes the following classroom features:

Traditional Lecture Method with Audio-Visual Aids

Only 1/16 portion of the courses at PeopleNTech is a traditional lecture course. Why? Simply because, on average, the weakest method for conveying IT knowledge and skills is the lecture method. Indeed, the only reason that PeopleNTech uses the lecture method at all is that there is still a small percentage of material of a general foundational nature that is best delivered through the lecture approach.

Demonstration

15/16 portion of the courses at PeopleNTech are conducted as actual demonstrations of how to do real-world IT tasks using the same actual tools that one encounters in the workplace. This method initiates and jump-starts learning by beginning with simple imitation, followed by repeated practice. This demonstration method ensures the opportunity to gain hands-on class practice, followed by additional exercises at home after class. This method facilitates optimum acquisition of the highest proficiency with the tools used. Additionally, students also have access to the recorded classroom presentation of all topics covered in class.

Hands-On Class Labs

The lab formula begins with a teacher demonstration, followed then during the same lesson by student replication of what the instructor has just shown to the class.

Tutoring

In order to graduate and receive certification from PeopleNTech, all students at the end of their course must tutor struggling less advanced students one on one for a certain number of hours. This tutoring simultaneously helps graduates to better understand the subject matter by forcing them to articulate it, while also giving struggling students extra help outside regular class time.

Student Public Speaking via Classroom Presentation Project

One curriculum requirement is that, for each of the courses that a student takes, he/she must prepare a formal oral presentation to “teach” a concept or procedure to a class of his/her peers. The student must also answer audience questions and defend his/her assertions. This process includes a formal evaluation and grade. In addition to helping each student to master the material at hand, this presentation requirement also develops a student’s public speaking skills, which are usually a requirement anyway for most mid-level jobs. Finally, the whole experience helps the student to prepare for tough questioning in job interviews.

2.4 Evaluation

As mentioned before, structured objective evaluation of student performance and social skills is the best measurement of IT skill training. The most effective evaluation is ongoing and summative. All such assessments are most useful if the

results can be evaluated against the set course objective(s) in order to determine the student's level of success or failure. PeopleNTech uses the following testing and sampling methods.

Assignments/Labs/Quizzes

Assignment/quizzes and/or labs are assigned to the trainees at the end of each class/topic so that the trainees get the opportunity to do brainstorming and practice with the technologies and tools and submit their work for showing their ability of working as good as in real-life work.

Class Test

Class tests are conducted at the end of each module of study for short modules. The instructors design it as per requirement for long modules. Qualifying in all class tests is a requirement of completion of the course.

Post-Course Boot Camp Lab

This is the equivalent of the students' final examination for their entire program prior to graduating and receiving a diploma. This entails completing a real-time project-based lab that covers the entire course curriculum.

Student Test Preparation Assessment Tools

As a PeopleNTech student, students can measure their acquired and retained knowledge by using our exclusive test preparation tools.

Vendor Exam Preparation

PeopleNTech recommends that its students, where appropriate, sit for their relevant vendor examination, and it puts such candidates in a simulated testing environment to facilitate full preparation for vendor tests.

2.5 Certificate/Diploma

PeopleNTech is authorized by the state authority to issue training diplomas after completion of all curricular requirements.

2.6 Post-Class Survey

A key element of PeopleNTech’s “quality control” is to measure student satisfaction with the overall learning experience and use the feedback to continuously improve training and services.

2.7 Top-Flight Job Placement Support

PeopleNTech does not merely train students. Indeed, an integral part of our overall service is providing actual job placement and career counseling within the IT field, which is part of why our institute manages to place 95% of its graduates in mid-level or senior-level IT positions within 4 months of graduation. All graduates receive the following:

Resume Assistance

Strong resume writing assistance is offered through occasional workshops.

Mock Interview Sessions

PeopleNTech’s comprehensive interview training prepares students for any kind of job. This process begins with a group lecture sharing tips and basic interviewing skills. Next, actual face-to-face mock interview role-playing sessions are conducted by PeopleNTech staff with authentic industry experience in hiring IT graduates; included are the following: candid evaluation of students’ interview performance, formulation of plans to improve interviewing skills, and advice on how to significantly boost the interviewee’s confidence and proficiency. Students become expert on key interviewing tips and techniques, mistakes people usually make, and how to formulate the best answers even to the most difficult questions in the most high-stress interview situations. Finally, this process serves to significantly improve students’ oral articulation skills.

Career Counseling and Job Placement Services

PeopleNTech is pleased to offer counseling by experienced IT industry professionals—not just an employment service. To this end, we offer both orientation sessions for new students and individualized services later on. With convenient and confidential career counseling, our institute’s trained counselors work with students one on one and via telephone to focus on immediate occupational

needs. In addition, counseling sessions cover access to job listings, employer contacts, and on-campus interviews. PeopleNTech recruiters give students the finest job placement services in the industry.

3 Belief Rule Base Approach

The assessment of skill level of students, achieved whether it is after the graduation from the universities or after getting skill by completing training program from PeopleNTech, is crucial. This will allow to reduce the gap between university education and the industry requirements. Consequently, appropriate teaching and learning methodology can be framed, allowing the fresh graduates to start their career at mid-level. The BRBES framework is demonstrated in Fig. 2, enabling to understand that the level of skill of a student depends on the factors mentioned in Fig. 1.

The reason for selecting BRBES approach is that most of the factors cannot be measured in a quantitative way because they are subjective in nature and hence inherit uncertainty. Therefore, without consideration of this uncertain phenomenon, the accuracy of the assessment cannot be achieved. This will in turn hamper the appropriate development of policy to embed skill to the employees of an organization. In addition, this model is flexible because it allows to add or delete any other factors. Moreover, it can easily be identified which factors should need to be given more priority than from the others because the weights or importance of each factor can be incorporated.

A BRBES includes two main components, namely knowledge base and inference engine [6]. Belief rules are used as the knowledge representation schema and Evidential Reasoning (ER) as the inference engine [7].

Belief rules are the extended form of traditional IF-THEN rules but equipped with belief structure to handle uncertainty [8]. A number of belief rules form Belief Rule Base (BRB), containing learning parameters such as attribute weight, rule weight, and belief degrees [9]. A belief rule is presented below:

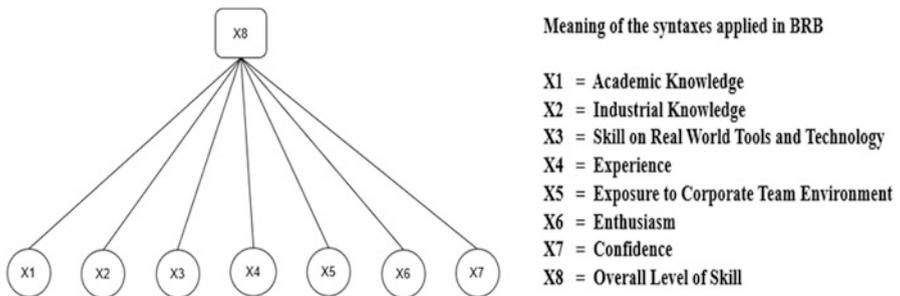


Fig. 2 BRB framework to evaluate overall level of skill

IF Academic Knowledge is High AND Industrial Knowledge is High AND Skill on Tools is Medium AND Experience is Low AND Team Environment is Medium AND Enthusiasm is Low AND Confidence is HIGH THEN Level of Skill is (High, 0.3), (Medium, 0.7), (Low, 0.0)

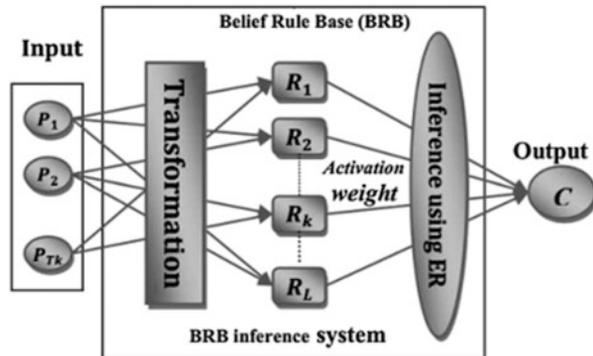
In this belief rule, the IF part consists of seven antecedent attributes, each with three referential values known as “High,” “Medium,” and “Low.” However, the THEN part consists of consequent attribute, namely “Level of Skill” with three referential values, which are embedded with belief degrees, thus forming a belief structure. This belief structure is complete because the summation of belief degrees is one ($0.3 + 0.7 + 0.0$). The belief structure is considered as incomplete when this summation is less than one causing due to ignorance or incompleteness [10]. In this way, uncertainty phenomenon is captured in the BRB. The relationship between antecedent and consequent attributes in a traditional IF-THEN rule is linear, but it is nonlinear in case of BRB [11]. Data obtained from interviews or surveys are usually nonlinear [12], and hence, this capability of BRB is found very effective. In this research, most of the data has been collected using interviews and surveys.

Evidential Reasoning (ER) can handle both qualitative and quantitative data [13]. Qualitative data inherently contains uncertainty. For example, the antecedent attributes that have been considered in the BRB framework (Fig. 2) are the examples of qualitative data. ER can process both qualitative and quantitative data in an integrated framework [14].

The inference procedures of BRBES consist of four steps, namely input transformation, rule activation weight calculation, belief degree update, and rule aggregation [15] as illustrated in Fig. 3.

The purpose of input transformation consists of distributing the value of an antecedent attribute over its various referential values [16]. This transformed value of the antecedent attribute, which is also known as input data, is termed as matching degrees to the referential values of an antecedent attribute [17]. When these matching degrees are assigned in a rule, it can be termed as packet antecedent, meaning that it becomes active [18]. For example, in the above rule, “Academic Knowledge” antecedent attribute is related to referential value “High.” However,

Fig. 3 Sequence of BRBES inference procedures



this antecedent attribute consists of three referential values “High,” “Medium,” and “Low.” The input data or the value of the antecedent attribute “Academic Knowledge” will be distributed over its three referential values, which could be (High, 0.6), (Medium, 0.2), and (Low, 0.2). However, the above rule only related to antecedent attribute “Academic Knowledge’s” referential value “High,” and hence “0.6” matching degree will be assigned to it. Since the number of rules of this BRBES will consist of 78,125 as it contains seven attributes each with five referential values, these matching degrees will be assigned to referential values associated with “Academic Knowledge.” Each of the 78,125 rules should contain “Academic Knowledge” antecedent attribute as well as any of its referential values. Thus, the above matching degree will be assigned to each of the 78,125 rules associated with “Academic Knowledge” attribute. In the same way, the matching degrees of the other antecedent attributes against their input values will be assigned. Hence, each of the 78,125 rules will be become active, and they are called packet antecedent.

However, these individual matching degrees of a rule against the antecedent attribute’s referential values should need to be combined [19, 20]. This is carried out by using weighted multiplicative equation, allowing the complementarity or the integration among the antecedent attributes of a rule [21, 22]. Eventually, the combined matching degree is used to calculate the degree of activation of a rule in the BRB. When the degree of activation of a rule is found zero, then the rule is considered as deactivated [23, 24]. The summation of the degree of activation of each of the 78,125 rules should be one.

There could be the case that the input data of any one of the antecedent attributes of the BRB framework cannot be acquired [25, 26]. This is example of uncertainty due to ignorance. In that case, the initial belief degrees that were assigned to each the 78,125 rules should need to be updated. This is called belief update [27, 28].

Finally, by using ER, all the activated rules are aggregated to obtain the output for the input data of the antecedent attributes [29, 30]. These output values are in fuzzy format, and they are converted into crisp value by using utility value against each referential value of the consequent attribute [31, 32].

4 BRBES to Evaluate Overall Level of Skill

This section presents the BRBES’s system architecture as well as its various components for evaluating skill level.

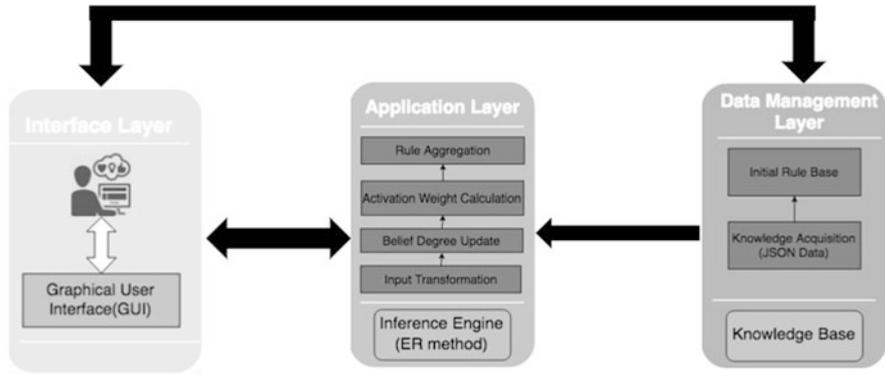


Fig. 4 BRBES architecture

Table 1 Referential values and utility values of antecedent attributes (VH, Very High; H, High; M, Medium; L, Low; VL, Very Low)

(a)

	Antecedent attributes																			
	x_1					x_2					x_3					x_4				
Referential values	VH	H	M	L	VL	VH	H	M	L	VL	VH	H	M	L	VL	VH	H	M	L	VL
Utility values	10	7	5	3	1	10	7	5	3	1	10	7	5	3	1	10	7	5	3	1

(b)

	Antecedent attributes														
	x_5					x_6					x_7				
Referential values	VH	H	M	L	VL	VH	H	M	L	VL	VH	H	M	L	VL
Utility values	10	7	5	3	1	10	7	5	3	1	10	7	5	3	1

4.1 Architecture

A three-layer architecture has been considered for the BRBES including data management layer, application layer, and Interface layer. Figure 4 illustrates the BRBES architecture.

Data Management Layer

Initial BRB is constructed in this layer, which is the knowledge base of the BRBES. The BRB framework as illustrated in Fig. 2 is considered to construct the knowledge base. Table 3 illustrates the initial BRB for the BRBES, while Tables 1 and 2 illustrate the utility values considered against each of the referential values related to the seven antecedent attributes and the consequent attribute (Table 3).

Table 2 Referential values and utility values of consequent attributes

	Consequent attributes				
	x_7				
Referential values	Very high	High	Medium	Low	Very low
Utility values	10	7	5	3	1

Table 3 Preliminary BRB for all rule base (VH, Very High; H, High; M, Medium; L, Low; VL, Very Low)

Rule ID	Rule weight	IF							THEN (x_8)				
		x_1	x_2	x_3	x_4	x_5	x_6	x_7	VH	H	M	L	VL
1	1	VH	1	0	0	0	0						
2	1	VH	VH	VH	VH	VH	VH	H	0.81	0.19	0	0	0
3	1	VH	VH	VH	VH	VH	VH	M	0.68	0.32	0	0	0
4	1	VH	VH	VH	VH	VH	VH	L	0.56	0.44	0	0	0
5	1	VH	VH	VH	VH	VH	VH	VL	0.43	0.57	0	0	0
...
78,121	1	VL	VL	VL	VL	VL	VL	VH	0	0	0	0.57	0.43
78,122	1	VL	VL	VL	VL	VL	VL	H	0	0	0	0.38	0.62
78,123	1	VL	VL	VL	VL	VL	VL	M	0	0	0	0.25	0.75
78,124	1	VL	VL	VL	VL	VL	VL	L	0	0	0	0.13	0.87
78,125	1	VL	0	0	0	0	1						

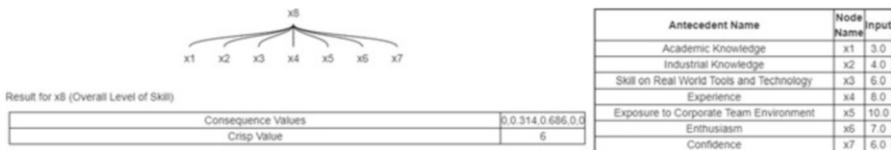


Fig. 5 BRBES interface to evaluate overall level of skill

Application Layer

Application layer comprises BRBES’s inference procedures, namely input transformation, rule activation weight calculation, belief update, and rule aggregation. The initial BRB of the data management layer is the input to the application layer and the inference procedures are on it.

Interface Layer

The interface layer provides a simple interface, enabling the production of BRBES’s output as shown in Fig. 5.

From Fig. 5, it can be seen that for the certain input values of the seven antecedent attributes ($x_1 = 3, x_2 = 4, x_3 = 6, x_4 = 8, x_5 = 10, x_6 = 7,$ and $x_7 = 6$),

the overall skill level (x_8) has been calculated in terms of both the fuzzy value (high = 0, medium = 0.314, and low = 0.686) and the crisp value (6). For this result, it can be opined that the assessment is complete because the summation of the referential values of consequent attributes is one.

However, these fuzzy values have been converted into a crisp value, which is “6” to obtain an overall view of assessment level of skill. By using this system, the policy maker will be able to identify the factors that are not performing well to have an impact on overall skill level. Consequently, appropriate decisions could be taken. In this way, the system allows the analysis of the problem from different perspectives.

5 Results and Discussion

The BRBES has been applied by collecting 200 data associated with the leaf nodes of its framework as illustrated in Fig. 2. For simplicity, Table 4 demonstrates only data of ten persons, where columns 2–8 show the data of the leaf nodes, while column 9 shows the BRBES created skill-level assessment results in terms of crisp value. Column 10 in Table 4 shows the expert assessment level of the skill.

Receiver operating characteristics curves (ROCs) are widely used to determine the accuracy of the prediction models. Therefore, an ROC has been considered as the method to calculate the prediction accuracy of the level of skill assessment carried out by the BRBES. Area under curve is considered as one of the important metrics in this method. When its value becomes one, then the accuracy of the prediction model like BRBES can be considered as 100%. SPSS 23 has been employed to generate the ROC curves as shown in Fig. 6, and the AUC data are illustrated in Table 5. Figure 6 illustrates the ROC curves allowing the comparison between skill-level assessment carried out by both BRBES and expert. An ROC curve having green line depicts

Table 4 Overall level of skill evaluation by BRBES and expert opinion

SL no.	x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	
								BRBES result	Expert opinion
1	8	6	6	6	6	6	6	6	4
2	5	7	4	1	10	5	3	5	5
3	3	4	6	8	10	7	6	6	5
4	7	7	7	9	9	8	6	7	6
5	6	9	8	2	9	4	7	6	5
6	2	7	7	5	4	5	5	5	4
7	4	7	5	4	6	3	1	4	5
8	4	1	2	2	10	2	8	4	4
9	2	3	2	1	6	4	10	4	3
10	3	8	7	5	7	10	9	6	5

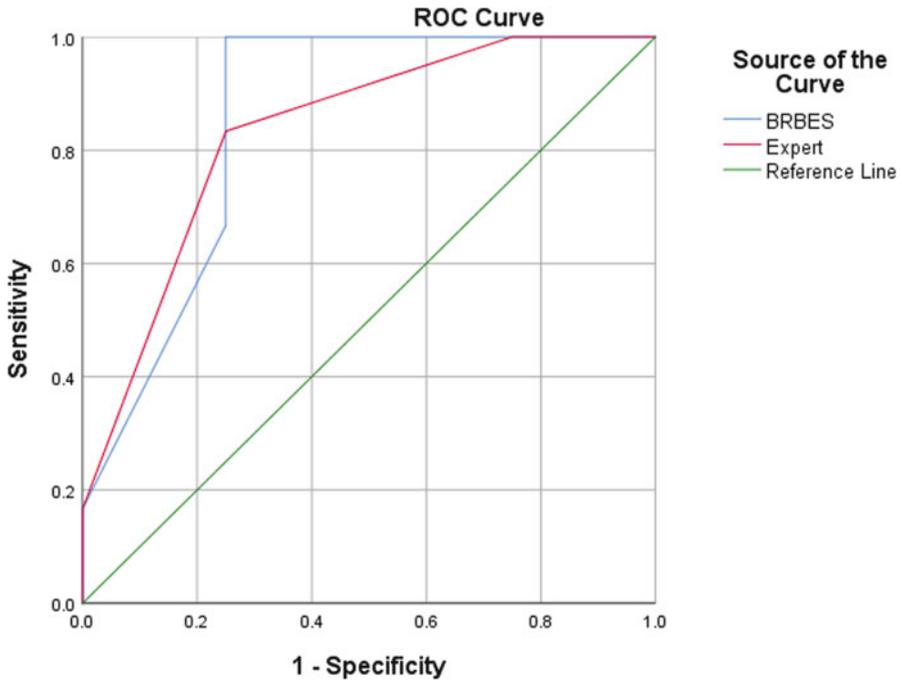


Fig. 6 ROC curves comparing BRBES’s result and human expert

Table 5 Comparison of AUC of BRBES and expert opinion

Test results variable(s)	Evaluated	Std. error	Asymptotic sig.	95% ACI	
	AUC			LB	UB
BRBES	0.854	0.144	0.070	0.571	1.000
Expert opinion	0.733	0.141	0.088	0.557	1.000

BRBES outputs while with gray line depicts expert opinion. Table 5 illustrates the AUC for BRBES and expert, which are 0.854 and 0.733, respectively. Therefore, it can be opined that the reliability of the skill-level assessment generated by BRBES is more dependable than that of expert opinion.

6 Conclusion

In order to best prove the efficacy of PeopleNTech’s IT training model, this chapter has utilized a Belief Rule-Based Expert System (BRBES). The beauty of BRBES is that it can evaluate even uncertain or abstract information and thus provide the most currently accurate prediction of IT graduates’ skills proficiency levels. Moreover,

BRBES actually outperforms the reliability of human experts because this system generates more accurate assessment metrics.

This chapter has presented a unique teaching and learning methodology that enables the new IT graduate to start professional life at mid-level positions. As we have seen, however, academic knowledge alone is not enough to obtain jobs of this caliber. Recent university graduates are generally unable to meet the requirements of mid-level jobs in the following senses:

- unfamiliarity with tools and technologies being used in today’s market,
- insufficient skill levels to satisfy market demand and current trends, and
- inadequate familiarity with the behavioral etiquette of corporate culture.

The root causes behind these proficiency shortfalls in the IT graduate population may be summarized as follows:

- the difficulty for universities and other educational institutions to achieve timely, responsive updates to their curricula in today’s global environment of rapidly changing information technology applications,
- market scarcity of closely cooperative industry–university relationships and/or lack of teacher/instructors with industry experience that is both current and sufficient in depth,
- students having inadequate opportunities for IT skills development in a real-world environment using current industry tools, and
- students’ lack of exposure to IT corporate culture.

This proprietary IT training model is a plausible solution for overcoming prevailing student limitations. It is also a welcome solution to meet industry’s current IT staffing needs.

References

1. R. Craig, *Employers Mistakenly Require Experience for Entry Level Jobs*. Forbes Magazine (2016)
2. J.H. Bishop, *The Incidence of and Payoff to Employer Training: A Review of the Literature with Recommendations for Policy* (1994)
3. R.H. Dave, Psychomotor levels, in ed. by R.J. Armstrong, *Developing and Writing Behavioral Objectives* (Educational Innovators Press, Tucson, 1970)
4. B.S. Bloom, *Taxonomy of Educational Objectives: The Classification of Educational Goals* (Cognitive domain, 1956)
5. D.R. Krathwohl, L.W. Anderson, *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives* (Longman, Harlow, 2009)
6. R. Karim, K. Andersson, M.S. Hossain, M.J. Uddin, M.P. Meah, A belief rule based expert system to assess clinical bronchopneumonia suspicion, in *2016 Future Technologies Conference (FTC)* (IEEE, Piscataway, 2016), pp. 655–660
7. M.S. Hossain, A.A. Monrat, M. Hasan, R. Karim, T.A. Bhuiyan, M.S. Khalid, A belief rule-based expert system to assess mental disorder under uncertainty, in *2016 5th International Conference on Informatics, Electronics and Vision (ICIEV)* (IEEE, Piscataway, 2016), pp. 1089–1094

8. M.S. Hossain, P.O. Zander, M.S. Kamal, L. Chowdhury, Belief-rule-based expert systems for evaluation of e-government: a case study. *Expert Syst.* **32**(5), 563–577 (2015)
9. S.T. Alharbi, M.S. Hossain, A.A. Monrat, A belief rule based expert system to assess autism under uncertainty, in *Proceedings of the World Congress on Engineering and Computer Science*, vol. 1 (2015)
10. M.S. Hossain, K. Andersson, S. Naznin, A belief rule based expert system to diagnose measles under uncertainty, in *World Congress in Computer Science, Computer Engineering, and Applied Computing (WORLDCOMP'15): The 2015 International Conference on Health Informatics and Medical Systems 27/07/2015-30/07/2015* (CSREA Press, London, 2015), pp. 17–23
11. M.S. Hossain, S. Akter, S. Rahaman, A belief rule based expert system to assess meditation, in *2015 International Conference on Computational Science and Computational Intelligence (CSCI)* (IEEE, Piscataway, 2015), pp. 829–832
12. R. Ul Islam, K. Andersson, M.S. Hossain, A web based belief rule based expert system to predict flood, in *Proceedings of the 17th International Conference on Information Integration and Web-Based Applications & Services* (ACM, New York, 2015)
13. T. Mahmud, K.N. Rahman, M.S. Hossain, Evaluation of job offers using the evidential reasoning approach. *Global J. Comput. Sci. Technol.* **13** (2013)
14. S. Rahaman, M.M. Islam, M.S. Hossain, A belief rule based clinical decision support system framework, in *2014 17th International Conference on Computer and Information Technology (ICCIT)* (IEEE, Piscataway, 2014), pp. 165–169
15. M.N. Jamil, M.S. Hossain, R. Ul Islam, K. Andersson, A belief rule based expert system for evaluating technological innovation capability of high-tech firms under uncertainty, in *Joint 2019 8th International Conference on Informatics, Electronics & Vision (ICIEV)* (IEEE, Piscataway, 2019)
16. M.S. Hossain, M.S. Khalid, S. Akter, S. Dey, A belief rule-based expert system to diagnose influenza, in *2014 9th International Forum on Strategic Technology (IFOST)* (IEEE, Piscataway, 2014), pp. 113–116
17. M.S. Hossain, E. Hossain, S. Khalid, M.A. Haque, A belief rule based (BRB) decision support system to assess clinical asthma suspicion, in *Scandinavian Conference on Health Informatics; August 22; 2014; Grimstad; Norway*, vol. 102 (Linköping University Electronic Press, Linköping, 2014), pp. 83–89
18. M.S. Hossain, I.B. Habib, K. Andersson, A belief rule based expert system to diagnose dengue fever under uncertainty, in *2017 Computing Conference* (IEEE, Piscataway, 2017), pp. 179–186
19. K. Andersson, M.S. Hossain, Smart risk assessment systems using belief-rule-based DSS and WSN technologies, in *2014 4th International Conference on Wireless Communications, Vehicular Technology, Information Theory and Aerospace & Electronic Systems (VITAE)* (IEEE, Piscataway, 2014), pp. 1–5
20. M. Hossain, M. Haque, R. Mustafa, R. Karim, H. Dey, M. Yusuf, An expert system to assist the diagnosis of ischemic heart disease. *Int. J. Integr. Care* **16**, A31 (2016)
21. M.S. Hossain, S. Rahaman, A.L. Kor, K. Andersson, C. Pattinson, A belief rule based expert system for datacenter PUE prediction under uncertainty. *IEEE Trans. Sustainable Comput.* **2**(2), 140–153 (2017)
22. M.S. Hossain, A. Al Hasan, S. Guha, K. Andersson, A belief rule based expert system to predict earthquake under uncertainty. *J. Wirel. Mobile Netw. Ubiquitous Comput. Dependable Appl.* **9**(2), 26–41 (2018)
23. S. Rahaman, M.S. Hossain, A belief rule based (BRB) system to assess asthma suspicion, in *16th International Conference on Computer and Information Technology* (IEEE, Piscataway, 2014), pp. 432–437
24. T. Uddin Ahmed, M.N. Jamil, M.S. Hossain, K. Andersson, M.S. Hossain, An integrated real-time deep learning and belief rule base intelligent system to assess facial expression under uncertainty, in *9th International Conference on Informatics, Electronics & Vision (ICIEV)* (IEEE Computer Society, Washington, 2020)

25. R.U. Islam, M.S. Hossain, K. Andersson, Inference and multi-level learning in a belief rule-based expert system to predict flooding, in *9th International Conference on Informatics, Electronics & Vision (ICIEV)* (2020)
26. R.U. Islam, M.S. Hossain, K. Andersson, A learning mechanism for BRBES using enhanced belief rule-based adaptive differential evolution, in *9th International Conference on Informatics, Electronics & Vision (ICIEV)* (2020)
27. M.S. Hossain, F. Ahmed, K. Andersson, et al., A belief rule based expert system to assess tuberculosis under uncertainty. *J. Med. Syst.* **41**(3), 43 (2017)
28. M.S. Hossain, Z. Sultana, L. Nahar, K. Andersson, An intelligent system to diagnose Chikungunya under uncertainty. *J. Wirel. Mobile Netw. Ubiquitous Comput. Dependable Appl.* **10**(2), 37–54 (2019)
29. T. Mahmud, M.S. Hossain, An evidential reasoning-based decision support system to support house hunting. *Int. J. Comput. Appl.* **57**(21), 51–58 (2012)
30. S. Kabir, R.U. Islam, M.S. Hossain, K. Andersson, An integrated approach of belief rule base and deep learning to predict air pollution. *Sensors* **20**(7), 1956 (2020)
31. M.S. Hossain, S. Rahaman, R. Mustafa, K. Andersson, A belief rule-based expert system to assess suspicion of acute coronary syndrome (ACS) under uncertainty. *Soft Comput.* **22**(22), 7571–7586 (2018)
32. M.S. Hossain, F. Tuj-Johora, K. Andersson, A belief rule based expert system to assess hypertension under uncertainty. *J. Int. Services Inf. Security* **9**(4), 18–38 (2019)

Innovative Methods of Teaching the Basic Control Course



L. Keviczky, T. Vámos, A. Benedek, R. Bars, J. Hetthéssy, Cs. Bányász,
and D. Sik

1 Introduction

System view, understanding systems and how they are controlled, is an important discipline in engineering education. Systems are all around us. Basic knowledge about them is important for everybody. Engineers need deep knowledge enabling analysis and design of control systems. Nowadays considering the ever-increasing knowledge, the explosion of information, the available visual technics and software tools, and the emerging requirement for online distance education, there is a need to revisit the content and the teaching methodology of the basic control course.

The basic control course held for software engineering students at the Budapest University of Technology and Economics in the spring semester 2019 covered the topics of analysis and design of continuous and discrete control systems. The content of the course and the teaching methods were overviewed to respond to the challenges of the new teaching environment. A new aspect in the content of the course is the introduction of the YOULA parameterized controller design, which is a very effective method. Other controller algorithms can be considered as special cases of YOULA parameterization.

Considering the teaching method, we tried to explain the main disciplines in an understandable way to everybody and then discuss the precise mathematical description. The developed multilevel e-book, SYSBOOK, also supported the

L. Keviczky (✉) · T. Vámos · Cs. Bányász
Institute for Computer Science and Control, SZTAKI, Budapest, Hungary
e-mail: keviczky@sztaki.hu; vamos@sztaki.hu; banyasz@sztaki.hu

A. Benedek · R. Bars · J. Hetthéssy · D. Sik
Budapest University of Technology and Economics, Budapest, Hungary
e-mail: benedek.a@eik.bme.hu; bars@aut.bme.hu; jhetthessy@aut.bme.hu;
siktdavid@gmail.com

understanding and provided some philosophical background to the different topics. Active learning deepens knowledge. Some interactive demonstrations presented during the lectures contributed to the joy of understanding. Active participation of the students was ensured by problem-solving at the end of the lectures and by the computer laboratory exercises using software MATLAB/SIMULINK. As Open Content Development, the students can contribute to the teaching material by elaborating their own case studies about a system and its control.

2 Content of the Basic Control Course

The course discusses analysis and design of both continuous and discrete linear control systems. Nowadays discrete control systems gain increasing importance in computer control of industrial processes. Control of linear, deterministic systems is discussed. To control a system, the model of the system should be first analyzed. Real systems are generally nonlinear, which can be handled individually. For analysis of linear systems, there are general methods. Therefore it is expedient to linearize the systems in a given environment of a working point and apply control methods using the linearized models of the systems. Input/output models and also state space models are used to describe the systems. The control system should ensure the required prescribed performance of the plant. The controller is designed considering the model of the plant and the quality specifications. Controller design methods are discussed both for input/output models and for state space models.

Eight lectures have been elaborated and are available in ppt form covering the following topics (<https://www.aut.bme.hu/Pages/ResearchEn/ControlTheory>):

1. Lecture: Introduction. Systems and control everywhere. Systems and their models. Analysis methods of continuous time linear systems.
2. Lecture: Analysis in the frequency domain. Relations between the time, Laplace operator, and frequency domain.
3. Lecture: Feedback control systems. Stability analysis. Quality specifications formulated in the time and in the frequency domain. Control structures improving disturbance rejection. *PID* controller design.
4. Lecture: State space representation.
5. Lecture: Controllability, observability, state feedback, state estimation.
6. Lecture: Sampled-data (discrete) control systems. Analysis in the time and in the *z*-operator domain.
7. Lecture: Description of discrete systems in the frequency domain. Relation to the continuous frequency functions. Discrete *PID* controller design. Discrete state equations. State feedback, state estimation.
8. Lecture: Control of discrete systems with time delay. YOULA parameterization. Smith predictor. Dead-beat control. Outlook.

For all lectures, problems are available for the students, which can also be reached on the above link. The students work on them at the last part of the lectures, and then get feedback about the solution and extra points for good solutions.

Recently published Springer textbooks [7, 8] support the learning process. We refer also to the textbook of Åström and Murray [2].

3 Method of Teaching the Basic Control Course

In the 3 hours of the lectures, 2 hours are devoted for lecturing and presentation; in the next hour, the students solve problems and then get immediate feedback of the solutions. During the semester, they have to work on a project designing continuous and discrete controller for a given plant. Besides these lectures, two problem-solving lectures prepare the students for the tests. Every second week, the students solve MATLAB/SIMULINK exercises in the computer laboratory with the guidance of the teacher.

Besides the presentations, visual interactive demonstrations have a convincing strength while providing also the joy of learning. Active problem-solving using software MATLAB/SIMULINK means learning by doing, while the students get some expert knowledge in analysis and design of control systems.

Laboratory work in the next semester is also very important. Use of distant or virtual laboratories would be also beneficial.

4 SYSBOOK Platform: Interactive Demonstrations

The idea of T. Vámos dating back for 20 years was to present the main principles governing systems and control on different levels, for everyone, for students, and for control experts [3, 11–13]. Nowadays there is an increasing demand to explain these concepts to everyone, simply and especially for non-engineering students [1].

System view could give a new dimension how to look at the phenomena around us. It could be useful for experts with nontechnical background as well. With system view in different areas of expertise, the systems could be better understood contributing to better decisions influencing their performance. Besides engineering, such areas are, for example, medicine, medical technics, economy, etc. System view is the basics of control engineering. New concepts in mathematical system theory could open new perspectives to modern areas of control design.

A multilevel e-book has been developed by T. Vámos and coworkers available at <http://sysbook.sztaki.hu/>. Its cover page is seen in Fig. 1.

The first level – knowledge for everyone (Fig. 2) – appears in cartoon-like form with explanations. The second level gives deeper explanations with mathematical descriptions for the students. Some animations and interactive files demonstrate the main concepts. Some pages are devoted to the third level dedicated for experts.

Fig. 1 Cover page of SYSBOOK

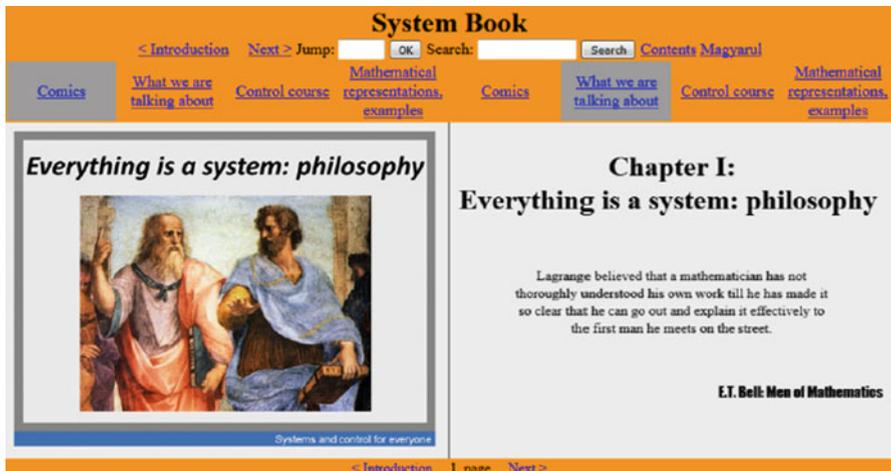
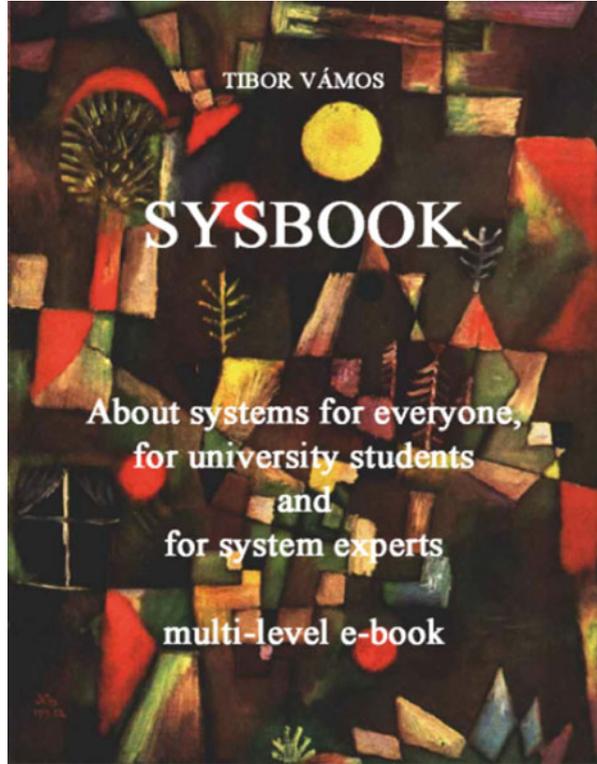


Fig. 2 Basic ideas can be explained for everyone. (Raffaello: The School of Athens, detail)

Fig. 3 Taking a shower may be a difficult process (Java applet)

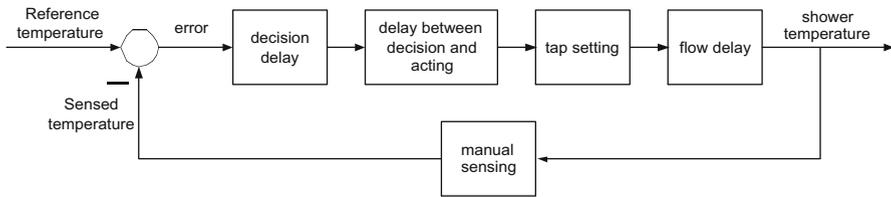
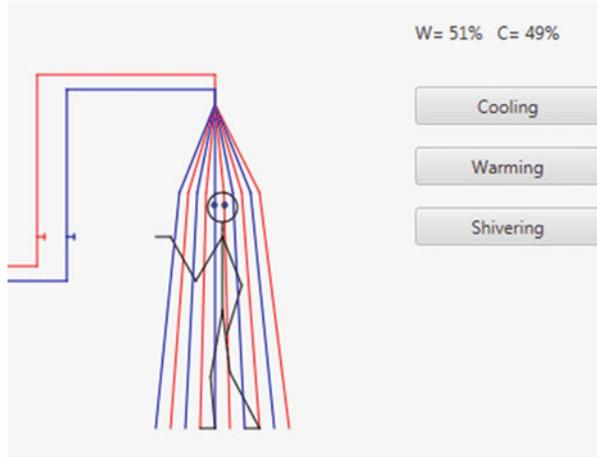


Fig. 4 Control is based on negative feedback

The pages of SYSBOOK appear on different surfaces that depend on the reader’s interest. Four categories are distinguished: comics; what we are talking about; control course; and mathematical representations and examples. Two surfaces do appear at the same time. The reader may navigate among them.

Case studies illustrate how system view and control disciplines can be applied in different areas (e.g., cooking, driving a car, energy production, oil refinery, some aspects of economy, systems and control in the human body, feedback in education, etc.).

SYSBOOK includes some demonstrative and interactive files visualizing system behavior. During the lectures, these parts of SYSBOOK can be used for demonstration. The case studies can be samples for the own work of the students.

As an example, Fig. 3 demonstrates that control of a system can be a difficult task. Taking a shower (Java applet) requires appropriate actions when changing the position of the taps considering the time delay of the process. Control is based on negative feedback, compares the measured output variable with its reference value, and uses the difference to modify the input variable of the system (Fig. 4).

If the action does not take into consideration the effect of flow delay, unstable performance could take place, and the temperature will change between hot and cold. By modeling the blocks and analyzing the behavior of the closed loop, some consequences can be drawn how to manipulate the system.



Fig. 5 Demonstration of the relationship between the time and the frequency domain

The behavior of systems can be analyzed in the time, frequency, and operator domain. The relationship between analysis in the time and in the frequency domain is illustrated by the interactive Java applet shown in Fig. 5. It presents that taking more sinusoidal components in the periodic input signal, the output of the system will be better approximated by the sum of the individual output components. The parameters of the system and the number of the sinusoidal components can be changed and the responses are visualized. So it is convincing that from the frequency response, consequences can be made for the time response of a system.

Another Java applet calculates and visualizes the step responses and frequency functions (Nyquist and Bode diagrams) of different systems.

Several control algorithms are discussed. Their behavior is demonstrated analyzing how the system tracks the reference signal, how it rejects the effect of the disturbances, how parameter uncertainties influence the performance, and what is the effect of the filters. Figure 6 demonstrates the behavior of a control system with *PID* controller. It is mentioned that MATLAB/SIMULINK ensures a better platform for controller design during the computer laboratories.

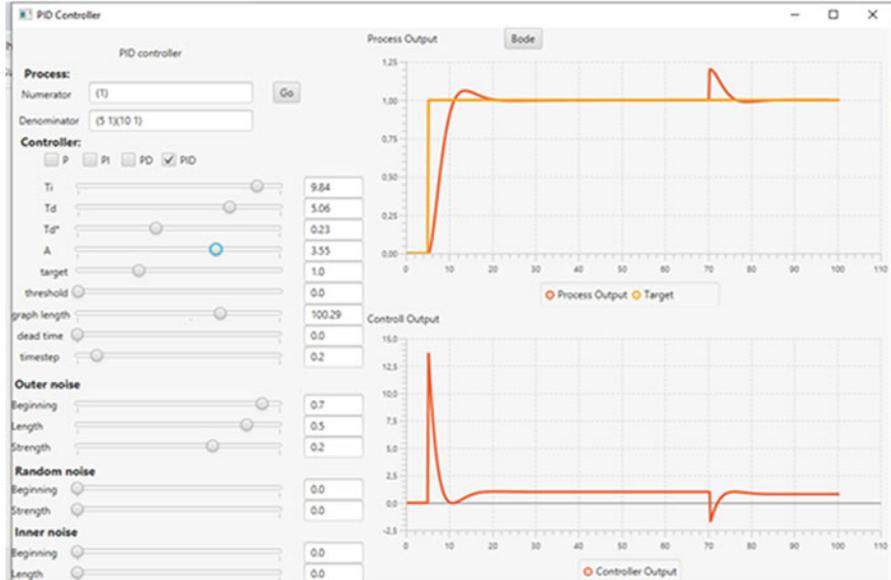


Fig. 6 Interactive Java file demonstrating the behavior of *PID* control

We refer here also to the interactive tools developed by Guzmán et al. [5, 6] for controller design.

5 New Paradigm in the Basic Control Course: Youla Parameterization

As a new feature, YOULA parameterization has been introduced as an essential control idea in the basic control course [9]. This approach follows from the basic feedback control idea and provides good properties for the control system especially in case of big dead time. The basic idea is shown in the sequel.

Control is based on negative feedback. In the theoretical part of the curriculum, properties of negative feedback are discussed. The block diagram of the control structure is shown in Fig. 7, where P is the model of the plant to be controlled, C is the algorithm of the controller, and F is the input filter.

This structure is effective ensuring reference signal tracking and disturbance rejection. The controller C is designed for the model of the plant considering the quality specifications. The most frequently applied algorithm is the *PID* controller.

Supposing a unity filter F an equivalent structure between the output y and the input r is given in Fig. 8. Q is called the YOULA parameter. Reference signal tracking would be ideal if controller Q would realize the inverse of the process model.

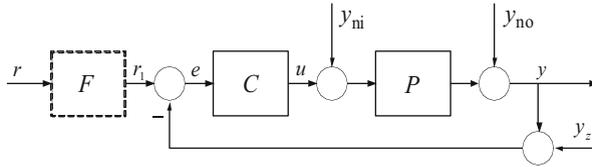


Fig. 7 Control is realized by negative feedback

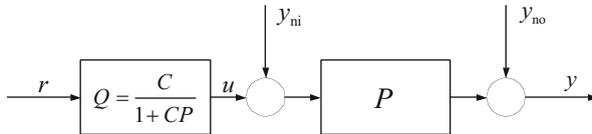


Fig. 8 Equivalent control structure with the YOULA parameter

Fig. 9 YOULA parameterized control with IMC

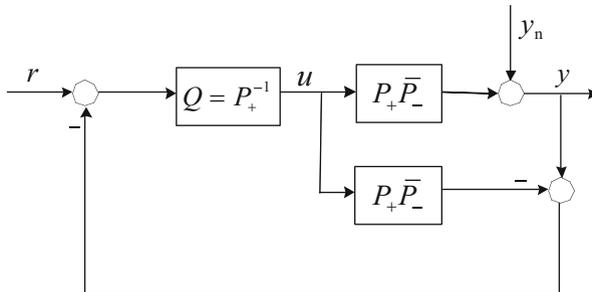
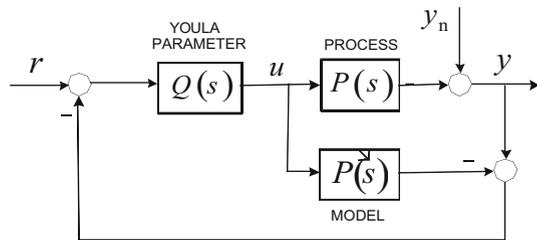


Fig. 10 Realizable YOULA parameterized control

But this structure cannot reject the effect of the disturbances. Therefore it is enhanced with Internal Model Control (IMC) [4] according to Fig. 9.

YOULA parameterized control can be used to control stable processes.

Generally the inverse of the process cannot be realized. The process model P should be separated to the invertible P_+ part whose poles can be cancelled and to the non-invertible part \bar{P}_- which contains the dead time and the non-cancellable poles. The YOULA parameter realizes the inverse of the invertible part of the process model (Fig. 10).

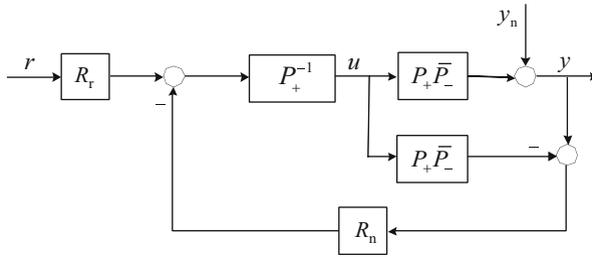


Fig. 11 YOULA parameterized control enhanced with filters

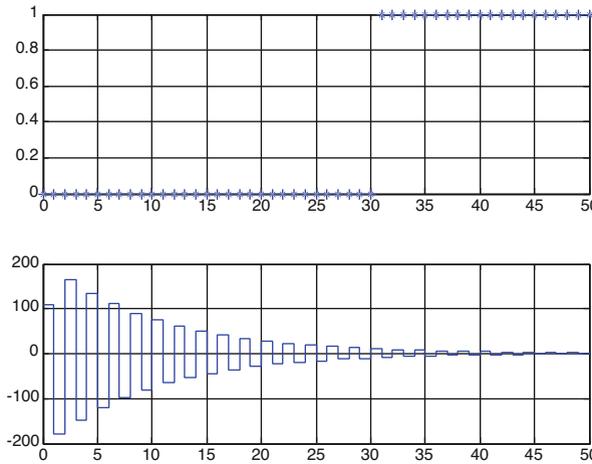


Fig. 12 Output and control signals when Q parameter cancels the whole dynamics

This structure can be enhanced by the R_r reference and R_n disturbance filters according to Fig. 11.

The role of the filters is threefold: the dynamics of reference signal tracking and disturbance rejection can be different ($2DF$ -two-degree-of-freedom controller), the maximum value of the control signal u can be restricted, and by appropriate choice of the filters, the control system can be made more robust, i.e., more insensitive to model uncertainties.

This structure can be applied both for continuous and discrete systems. For discrete systems, instead of the transfer function P the pulse transfer function G is used.

Figure 12 shows the output and control signals for control of a discrete second-order system with big dead time when cancelling the whole dynamics. Oscillations in the control signal will cause intersampling oscillations in the output signal, while reaching the required reference value in the sampling points. Figure 13 gives the

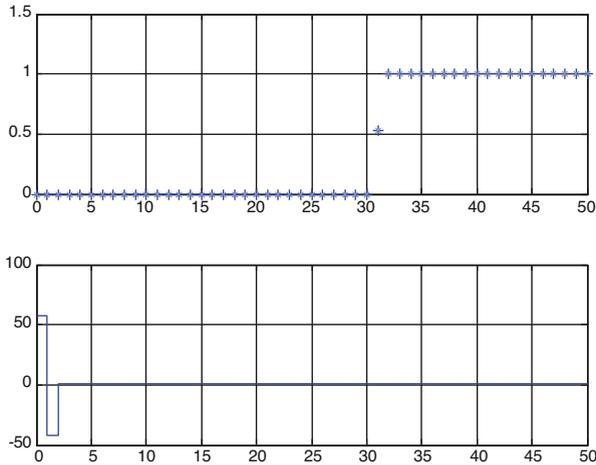


Fig. 13 Output and control signals when Q parameter cancels only the invertible part of the model

signals when only the invertible part is inverted in the controller. It is seen that the control performance became calm. Here filters were not applied.

The YOULA parameterized algorithm is especially effective when the process contains big dead time, and with appropriate design of the filters it is less sensitive to parameter uncertainties than the other algorithms.

It is shown that other control algorithms as *PID*, Smith predictor, and dead-beat control can be considered as special cases of the YOULA parameterized algorithm.

6 MATLAB/SIMULINK Computer Exercises

The basic software applied in the computer laboratories is MATLAB/SIMULINK. The students get some expertise in applying analysis and synthesis methods in problem-solving.

A MATLAB exercise description gives a short summary of the considered topic and then provides the examples from the simplest to the more complex ones. A MATLAB exercise related to a given topic can be executed within a 2 hour time frame and provides the knowledge for further individual work in the given topic. The students generally do not get a ready program; they have to build it command by command. This way, they have to think over and understand the analysis or design procedure step-by-step. Based on these exercises, the students are prepared to solve basic control problems and to solve the homework project. The MATLAB exercises cover the following topics: Introduction to MATLAB/SIMULINK and to the control toolbox. Properties and characteristics of typical control elements in the time and in the frequency domain. Stability analysis. *PID* controller design. State

space description. Controllability, observability. State feedback, state estimation. Sampled data control systems. Z-transform and pulse transfer functions. Controller design based on the YOULA parameterization. Discrete *PID* controller design. Smith predictor. Dead-beat control.

Generally the problem is solved using MATLAB, and then a SIMULINK program is built to simulate the behavior of the control system. In some cases a core program is given for a specific problem, and the students give the input data and running the program they evaluate the behavior of the system.

As an example, the core program of the discrete YOULA parameterization is presented in the sequel.

```
% Youla_discrete basic program
Q=minreal(Rn/Gp,0.0001)
C=minreal(Q/(1-Q*G),0.0001)
L=minreal(C*G,0.0001)
Tr=minreal((Rr/Rn)*Q*G,0.0001)
Ur=minreal((Rr/Rn)*Q,0.0001)
t=0:Ts:50;
yr=step(Tr,t);
subplot(211), plot(t,yr,'*'),grid
ur=step(Ur,t);
subplot(212), stairs(t,ur),grid
```

Then the user defines the process (e.g. second-order system), gives the sampling time, calculates the pulse transfer function and gives its separation to invertable and non-invertable parts, and gives the filters. The MATLAB code is:

```
clear; clc; s=zpk('s')
P=1/((1+5*s)*(1+10*s))
Ts=1; z=zpk('z',Ts); G1=c2d(P,Ts)
G=G1*z^(-30)
Gm=1; %G-
Gp=G1/Gm %G+
Rr=1/z; Rn=1/z;
```

Then call the program. The behavior of the algorithm can be investigated with different separation and with different filters. The program can be enhanced by calculating the responses for the disturbances as well. The SIMULINK diagram can be built (Fig. 14) enabling analysis of intersampling behavior as well.

The MATLAB exercises supporting the control course are given in Keviczky et al. [8]. The chapters of the exercise book are fitted to the chapters of the theoretical material.

7 Open Content Development: Student Case Studies

Nowadays in education a new teaching-learning paradigm is Open Content Development (OCD) which means active participation of the teachers and students creating an up-to-date teaching material. This project runs at the Department of

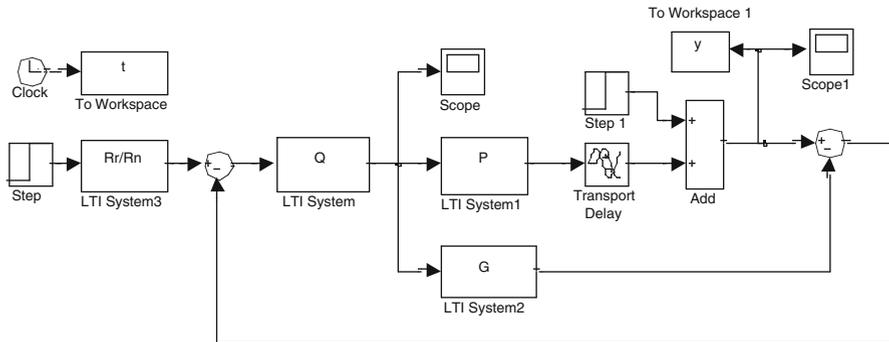


Fig. 14 SIMULINK program for the YOULA parameterized control system

Technical Education at the Budapest University of Technology and Economics since 2015 supported by the Hungarian Academy of Sciences [3]. In the frame of vocational teacher training programs, several so-called micro-contents have been developed. Utilizing the experiences of these pilot efforts, this approach seemed to be fruitful also in basic control education.

The SYSBOOK platform has been connected to the OCD model. SYSBOOK provides several case studies for systems and their control. Teachers and students studying systems and control can elaborate new case studies in their areas of interest which means active application of the learned topics. After evaluation these projects can be uploaded in the student area of SYSBOOK.

The system chosen by the students is modeled. The control tasks are formulated. In the different examples it is investigated, what is the considered system, how is the system connected to its environment, what are its input signals, and what are its output signals? What happens between them? How can this be described mathematically? What are the requirements set for the system? Can we control the system? How to control the system?

Some uploaded student projects are temperature control of a terrarium, speed control, model of the blood circulation and the respiratory system, the model of building a house, etc. (Fig. 15). Every semester, the student area is supplemented by new case studies.

The system open for the participating students/learners and educators is accessible through the research web page (www.oed.bme.hu); the page also contains bring your own device (BYOD) approaches that serve the methodological support of the innovations implemented within the open system.

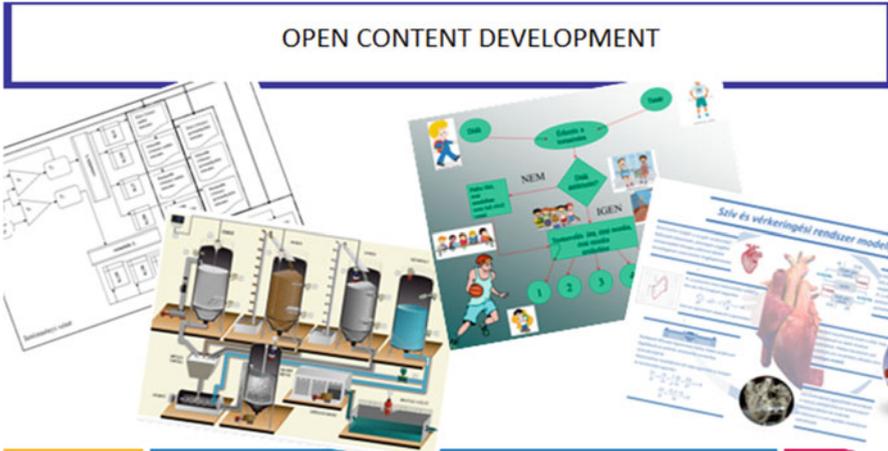


Fig. 15 Some student projects

8 Conclusions

Nowadays considering the ever-increasing knowledge, the explosion of information available at the Internet, the available visual technics and software tools, and the requirements for online education, there is a need to revisit the content and the teaching methodology of the basic control course.

Eight lectures have been elaborated which are available for teaching. Corresponding problems with the solutions are also provided.

As a new paradigm, controller design based on YOULA parameterization has been introduced already in the first control course. It is shown that some other control algorithms can be considered as special cases of YOULA parameterization.

In the methodology of teaching, a basic control course the motivation of the students can be increased by active participation in the learning process, including interactive demonstration of the principles, solving exercises at the end of the lectures, solving analysis and synthesis problems in the computer laboratories, and developing own case studies for SYSBOOK in OCD framework.

It should be also emphasized that the examples of systems and their control should be chosen mainly from the area of the specialization of the students (electrical or software engineering, chemical engineering, biology, economics, etc. [10]). Also it is important to provide real-time experiments in laboratory work or using distant laboratories. IFAC Repository would be also of great help reaching useful resources.

References

1. P. Albertos, I. Mareels, *Feedback and Control for Everyone* (Springer, 2010)
2. K.J. Åström, R.M. Murray, *Feedback Systems: An Introduction for Scientists and Engineers* (Princeton University Press, 2008), http://www.cds.caltech.edu/~murray/books/AM05/pdf/am08-complete_22Feb09.pdf
3. A. Benedek, T. Vámos, R. Bars, D. Sik, *Open content development applied in learning systems and control* (European Control Conference (ECC), Napoli, 2019), pp. 3059–3064
4. C.E. Garcia, M. Morari, Internal model control 1. A unifying review and some new results. *Indust. Eng. Chem. Proc. Des. Develop.* **21**(2), 308–323 (1982)
5. J.L. Guzmán, K.J. Åström, S. Dormido, T. Hägglund, Y. Piquet, Interactive learning modules for PID control. *IFAC Proc. Vol.* **39**(6), 7–12 (2006)
6. J.L. Guzmán, T. Hägglund, K.J. Åström, S. Dormido, M. Berenguel, Y. Piquet, *Understanding PID Design Through Interactive Tools, 19th IFAC World Congress* (Cape Town, 2014)
7. L. Keviczky, R. Bars, J. Hetthéssy, C. Bányász, *Control Engineering* (Springer, 2019a)
8. L. Keviczky, R. Bars, J. Hetthéssy, C. Bányász, *Control Engineering: MATLAB Exercises* (Springer, 2019b)
9. L. Keviczky, C. Bányász, *Two-Degree-of-Freedom Control Systems, The Youla Parameterization Approach* (Academic Press, Elsevier, 2015)
10. A. Leva, Teaching PID control to computer engineers: a step to fill a cultural gap. 11th IFAC Symp. Adv. Cont. Educ., ACE'2016, Bratislava, Slovakia, *IFAC-Papers Online* **51**(4), 328–333 (2016)
11. T. Vámos, J. Bokor, K. Hangos, Systems - governing principles and multimedia /CD/, in *14th IFAC World Congress*, (Beijing, 1999), PT-5, p. 79. Plenary lecture
12. T. Vámos, R. Bars, D. Sik, Bird's eye view on systems and control – General view and case studies. 11th IFAC Symp. Adv. Cont. Educ., ACE'2016, Bratislava, Slovakia, *IFAC-PapersOnline* **49**(6), 274–279 (2016)
13. T. Vámos, L. Keviczky, R. Bars, A. Benedek, D. Sik, An introductory overview about systems and control: a motivation lecture in control education, in *26th Mediterranean Conference on Control and Automation (MED'2018)*, (Zadar, 2018)

Part III
**Frontiers in Education – Methodologies,
Student Academic Preparation and
Related Findings**

Towards Equitable Hiring Practices for Engineering Education Institutions: An Individual-Based Simulation Model



Marcia R. Friesen and Robert D. McLeod

1 Introduction

This work presents developed computer modelling and simulation to investigate the relative impacts of various hiring strategies of underrepresented groups (URGs) in the professoriate of engineering schools at Canadian universities. Hiring practices typically represent a key component of the overall equity, diversity, and inclusion (EDI) strategy in these schools, and EDI in STEM fields are presently very active areas of discussion and policy implementation generally.

In the engineering profession in Canada including academia, women and Indigenous persons are likely the two greatest URGs, and men are likely the greatest overrepresented group (ORG). Underrepresentation in engineering studies is impacted by family and community of origin, K-12 schooling, and popular culture. Similarly, engineering practice both inside and outside academia holds a myriad of experiences and structural features that can either enable or deter URG participation and persistence. For example, studies show that women in engineering persist when they feel a sense of belonging in their workplace, feel confident in their technical abilities, and are supported by a respectful and equitable work environment. Studies also show that women primarily leave due to an unwelcoming culture and job inflexibility that leads to sacrifices with respect to work-life balance, although lack of opportunity for advancement can be a contributing factor [1–8]. However, prior to any of the above elements occurring, systemic implicit or unconscious bias (against women, people of colour, Indigenous Canadians, and others) can limit URG access to professional opportunity in the first place.

M. R. Friesen (✉) · R. D. McLeod

Electrical and Computer Engineering, University of Manitoba, Winnipeg, MB, Canada
e-mail: marcia.friesen@umanitoba.ca; robert.mcleod@umanitoba.ca

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_18

265

A research basis exists for the benefits of a diverse professional workforce, including improved financial performance of organizations (revenues, equity, operating profit, and/or stock price), responsiveness to diverse markets seen in improved quality and customer satisfaction, increased employee engagement and productivity, better decision-making, and reduced legal and reputational costs [9–13].

Conversely, one can also develop the case for EDI by examining where EDI has failed. For example, in the development of the automotive crash test dummy, the initial design, ‘Sierra Sam’, was a 95th percentile male dummy. Subsequently, 50th percentile dummies were modelled after an average male in height, mass, and proportion. Decades later, a female dummy was added that represented a 5th percentile Caucasian woman at a diminutive 152 cm (5 ft) tall and 50 kg (110 lb). Ill-fitting personal protective equipment (PPE) is often a result of design based on the body and head proportions and shapes of Caucasian male populations, with statistically significant impacts on health and injury including death [14].

One could argue that the engineering faculty hiring process suffers from similar design flaws, and EDI discussions in engineering schools in Canada often look at faculty hiring committees as one necessary point of intervention. Hiring practices in academia have a few characteristics that are not necessarily shared across all industries. Academic hires are often for life; therefore, the time constant on change is high, but the attrition rate is also predictable. Second, for many years already, the supply of applicants with the basic qualifications for academic positions has far exceeded the number of available positions. Third, the range of the academic role of teaching, research, and service means that there is no one benchmark for ‘qualified’ or ‘best’ candidate. Rather than comparing apples to apples, one is often comparing an appealing apple to an appealing cheese.

While EDI is absolutely a field with many qualitative, layered, and complex interacting factors that are best understood qualitatively than quantitatively, it is also true that a quantitative approach to a focused component of the issue (professorial hiring in engineering) has the potential to be insightful and its results will appeal to data-driven individuals such as engineers, who may in fact subconsciously dismiss other EDI initiatives and data on their qualitative basis.

2 Objective

This work is a simulation study of hiring policy, to investigate the relative impacts of various hiring strategies of underrepresented groups in academia. In the simulations that follow, there is an implicit assumption that EDI is a desirable objective to pursue. The underlying algorithms that comprise the model combine both deterministic elements (i.e. specific interventions) with stochastic elements that are, as a consequence, probabilistic in nature (e.g. probability of equally qualified persons being hired out of short-listed groups).

3 Modelling and Simulation

A modelling and simulation paradigm that is well suited to these types of thought experiments is generally denoted agent-based modelling (ABM) and simulation (ABMS). ABM is ‘bottom-up’ systems modelling from the perspective of constituent parts. Systems are modelled as a collection of agents (in social systems, most often people) imbued with properties: characteristics, behaviours (actions), and interactions that attempt to capture actual properties of individuals. In the most general context, agents are both adaptive and autonomous entities who are able to assess their situation, make decisions, compete or cooperate with one another on the basis of a set of rules, and adapt future behaviours on the basis of past interactions. In this work, a simpler variant is used as the agents are simply individuals from different groups without any additional agency or interactions.

A key advantage of ABM is the ability to model interventions or policy directions that are either too risky, costly, and/or too long in duration to test in real life. Further, the foundational premise and the conceptual depth of ABM is that simple rules of individual behaviour will aggregate to illuminate complex and/or emergent group-level phenomena *that are not specifically encoded by the modeller*. Emergent behaviour may be counterintuitive or surprising and may be a simple or complex behavioural whole that is greater than the sum of its parts. The ability of system-level outcomes to elude simple prediction based on the known rules that govern agent behaviour is a cornerstone of emergence [15–17].

Baseline Assumptions A majority group (ORG) is labelled Group A and a minority group (URG) is labelled Group B. No attribution is given to Group B other than being the URG, which may be due to gender, racialized persons, socio-economic class, or other characteristics. However, in a school or college of engineering, it can be largely inferred to be gender.

To start, the organization has roughly 90 Group A and 10 Group B individuals in their existing workforce (i.e. 10%). The hiring process model is modelled as one new individual every 3 months. The attrition rate was one per year, weighted in proportion to the percentage of Group A and Group B individuals in the organization.

An assumption is that the organization has agreed that Group B is clearly underrepresented and should be, at minimum, 25% of the overall workforce. We have selected 25% as opposed to other targets since quarters are easy to visualize and discuss; this target is adjustable as well. A further assumption is that the organization agrees that limiting position postings and hiring solely to Group B candidates would not be fair or equitable to either Group A or Group B.

Corner cases Corner cases are a first approach to establishing validity of any simulation model. Two such cases were investigated here. In the first instance, a simulation was run to typify today’s hiring processes which claim to be URG- or gender-‘blind’ and only interested in the ‘best’ candidates, i.e. no interventions of any type were considered for either Group A or Group B specifically. Every

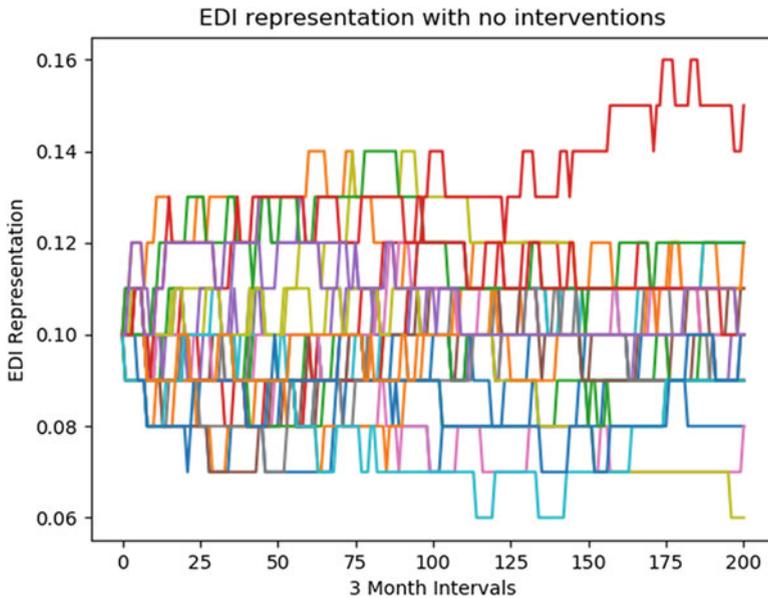


Fig. 1 EDI representation trajectories with no EDI incentives or interventions (status quo)

applicant is ranked and sampled from a uniform distribution. As per Fig. 1, there is essentially no change to Group B representation over time in this status quo approach. The average value ('qualification score') of the persons hired was approximately 99.3%. Applying macro face validation would affirm this result, namely, that the implication of no change is to see no change. The expected number of applications that a person submits for a faculty position before being hired is roughly four for the best overall candidate from Group A.

The status quo scenario (Fig. 1) does not imply that there is no institutional EDI initiative in place, such as the inclusion of a boilerplate university-wide EDI policy within each position posting, which have limited impact on their own. Figure 1 also demonstrates that in the status quo scenario, some trajectories nonetheless give the impression of improving the URG's representation over time. For example, the red (top) trajectory is just as statistically probable as the worst trajectory in the set of simulations but could nonetheless lead to administrators taking credit for a statistical anomaly.

A second corner case simulation considers only hiring Group B applicants going forward. Group B applicants are considered and ranked from a uniform distribution. As per Fig. 2 and intuitively expected, this leads to a continually increasing representation of Group B to the target and beyond. The average value ('qualification score') of the persons hired was approximately 91.8%. Applying macro face validation would affirm this result, that is, if you only select candidates from Group B, eventually everyone will be Group B. As the application pool is

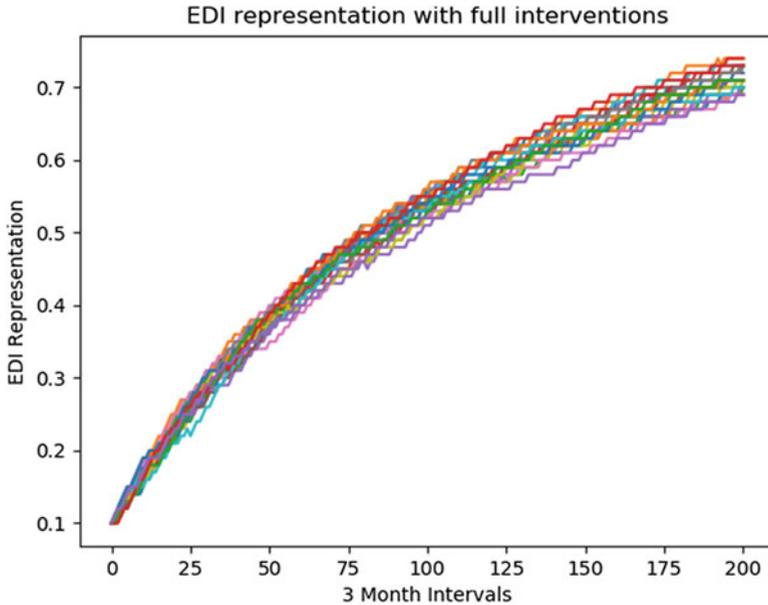


Fig. 2 EDI representation trajectories with full EDI incentives (hire only Group B)

smaller than the previous corner case ('status quo'), the probability of hiring the overall best irrespective of group is reduced while still very respectable overall.

Subsequent to corner case validation, several scenarios were investigated.

Scenario 1 Assume a position is posted, for example, for an Assistant Professor in Electrical and Computer Engineering. As is often the case, it is not unreasonable to receive 105 applications. Of these, it is usually fairly easy to identify applicants as being in either Group A or Group B and assume five are from Group B. The actual starting numbers in Group A or B are not critical except that they be different from the target outcome.

At this point, an algorithmic approach is introduced: select all five Group B applicants and uniformly sample the remaining 100 applicants to randomly select 15 of 100. The new group of candidates for the position are the five from Group B and 15 (sampled) from Group A. These proportions correspond to the target outcomes (25% from Group B).

A search committee reviews the combined pool of 20 applicants and shortlists the four applicants of interviews, where the shortlist is composed of the top two candidates from each group (an additional intervention). Probability statistics tell us that within this group, one likely has a top 6–7% candidate of the entire applicant pool, and they would be from either Group A or Group B. This is arguably as good a candidate as one might expect in any hiring competition.

This approach may be coupled with earlier interventions with Group B applicants that may arguably make them, on average, more qualified against the position requirements than Group A candidates. This corresponds to actively ‘shoulder tapping’ highly qualified Group B candidates to encourage them to apply to increase the average qualification score of Group B applicants. Assuming the average ‘qualification score’ (for the given position) of Group A is 50 and the average ‘qualification score’ of Group B is 75 (averages of uniform distributions 0–100 and 50–100, respectively), the expected value of the best from Group A would have a rank of 93.75, while the expected best of Group B would be 91.66.

While there are always limitations to an algorithmic approach and its underlying assumptions, it is worth considering when compared to no strategy at all. To the anticipated argument that with uniform sampling of 15 applicants from Group A, one is likely not getting the best possible candidate, the simulations demonstrate that the expected best ranking of a sample of 15 is still in the nineties. While probability and statistics defend this approach, anecdotal experience does as well. There are many near the top, and there is often little to differentiate those in the top 10% or so within any pool of applicants.

A benefit of this scenario is that in the steady state, everyone is satisfied with respect to the overall target goal. In the long run, no one should feel as though they obtained the job just because they were in Group B, nor can a Group A individual believe they were not given a leg up as they were unlikely (statistically) the best overall. In the event that no applicants are considered hireable, the position can be reposted, inviting those from Group A who were not selected in the sampling process (Fig. 3).

Substantive gains can be seen in the simulation results illustrated in Fig. 1 as Group B representation increases from 10% to near 20% in 10 years and increasing over time towards the target (with noted variability between simulation runs). Yet, although the selection process appears heavily weighted to Group B applicants, it nonetheless takes roughly 40 years to reach the representation target.

The average value (‘qualification score’) of the persons hired was approximately 95.5%. The expected number of applications the best candidate overall is expected to apply for is 100/15 or 7 positions in this simple model. This is only a handful of applications to get to an interview and is not unlike the number of applications a candidate will send out when seeking a faculty position.

Scenario 2 Another scenario to consider is one in which the application pool or Group B grows through incentives, early recruitment into undergraduate programs through scholarship such that the application pool grows towards 25% within 25 years. Trajectories of EDI representation in this case are illustrated in Fig. 4. As with any scenario simulated, the timeframe to achieve any reasonable targets appears excessive. To illustrate this point further, Fig. 5 extends the simulation to 200 years. A similar potential criticism is that for the best overall candidate from Group A, the expected number of applications to be hired into a faculty position is roughly 12. The expected value (‘qualification score’) of the hire is approximately 99.

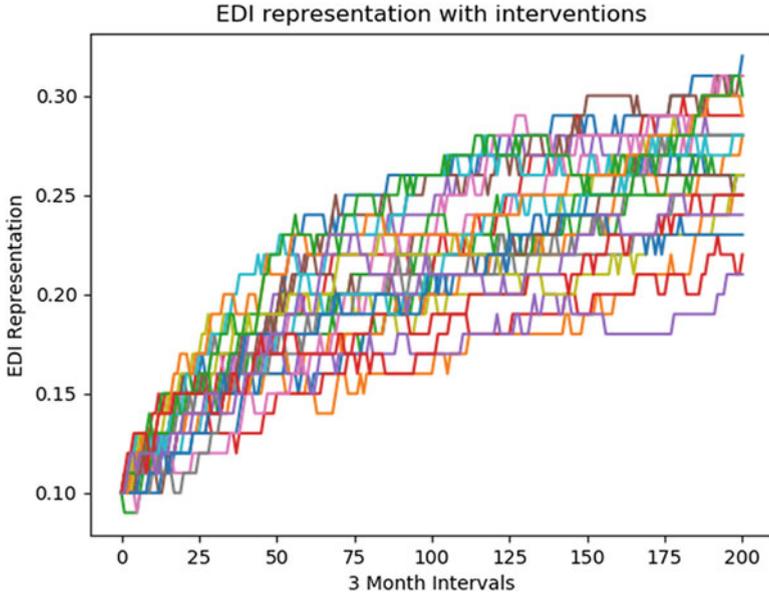


Fig. 3 EDI representation trajectories with interventions (Scenario 1)

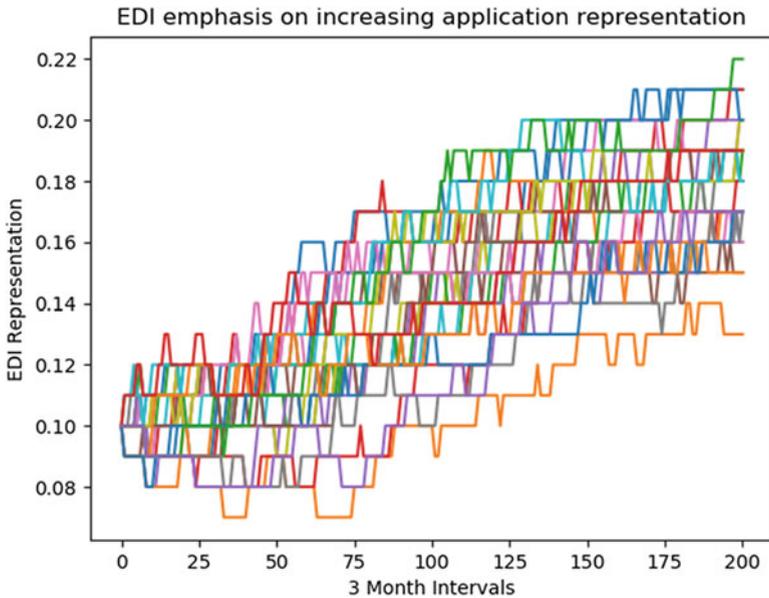


Fig. 4 EDI representation trajectories with applicant pool at the desired or expected representation (Scenario 2)

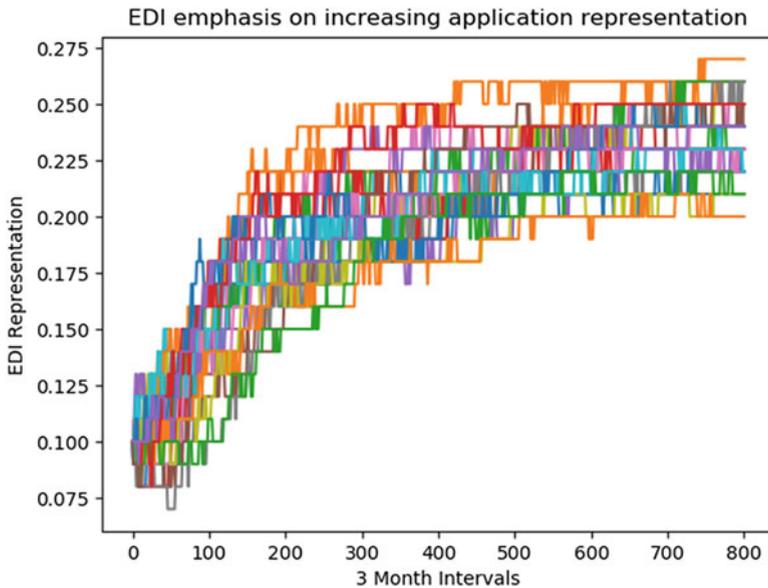


Fig. 5 EDI representation trajectories applicant pool at the desired or expected representation extended to 200 years (Scenario 2)

Scenario 3 Another scenario may be to consider full or aggressive EDI incentives followed by a period of ‘status quo’ hiring, assuming the applicant pool has achieved desired representation. For this simulation, full EDI intervention was in place for 6 years, followed by an assumption that the application pool reached the representative percentage of the target. This is illustrated in Fig. 6, where there is a Group B hiring only, followed by ‘blind’ or ‘best candidate’ hiring, assuming both the URG representation in the faculty and ongoing application pool reached its target of 25%. The rationalization may be that once an institution is seen as having an aggressive EDI policy, more URGs may be inclined to apply in the future. However, the simulation results demonstrate that the gains are not sustained, and Group B representation regresses over time in ostensibly ‘blind’ or ‘best candidate’ processes, where if the processes are truly ‘blind’ or ‘best candidate’, the Group B representation would remain constant at the target percentage.

Scenario 4 Fig. 7 illustrates a similar scenario of aggressive EDI interventions for 6 years, followed by the Group B applicant pool only reaching 15%, followed by ‘blind’ or ‘best candidate’ hiring.

Summary statistics of the simulations described above are presented in Table 1.

These results challenge the argument that one often hears that being that actively working towards diversity in hiring will be at the expense of excellence or quality. The average value (‘qualification score’) shown in Table 1 combined with the diversity of ways that academic work can be considered meritorious supports

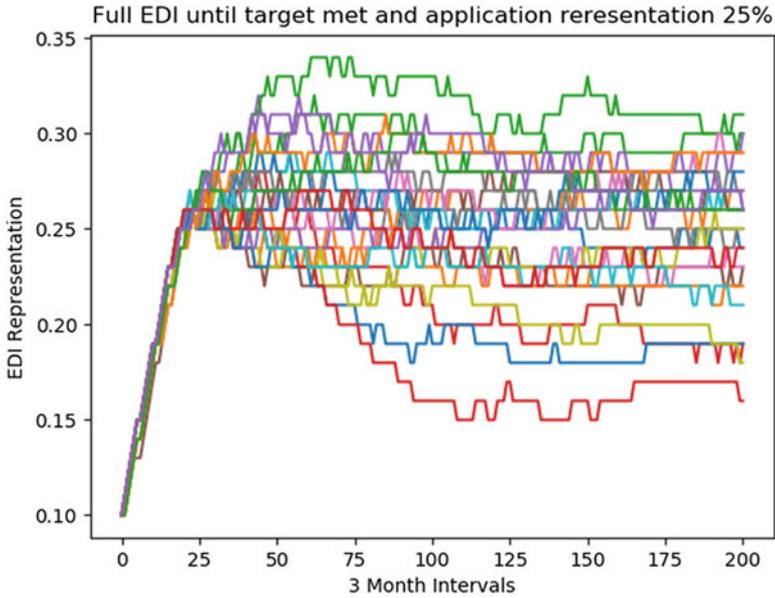


Fig. 6 Representation trajectories with full EDI (to 25% representation) followed by selecting the 'best applicant' (status quo) (Scenario 3)

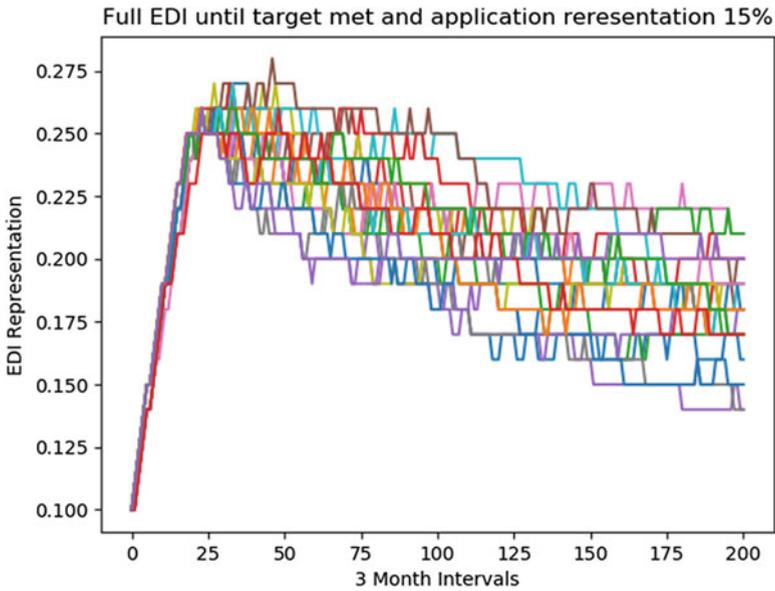


Fig. 7 Representation trajectories with full EDI (to 25% representation), followed by selecting the 'best applicant' (status quo) with Group B applicant pool representation at 15% (Scenario 4)

Table 1 Summary of model statistics

Scenario	Years to 25% Target	Average value ('qualification score') of hires	# of applications for 'best' Group A
Corner case 1	Never	99.3	<4
Corner case 2	6.25	91.8	n/a
Scenario 1	40+	95.5	28
Scenario 2	200	99	12
Scenario 3	6.25	98	n/a

Table 2 Variations on Scenario 1

Scenario 1 variations	Years to 25% Target	Average value ('qualification score') of hires	# of applications for 'best' Group A	30 by 30 target (%)
Shortlist 2 Grp A/2 Grp B	40+	95.5	28	37
Shortlist 2 Grp A/1 Grp B, with Grp B weighted, on average, more highly qualified	50-	95.6	18	31
Shortlist 2 Grp A/1 Grp B	50+	98.8	3.7	30
'Best overall' with increasing Grp B in applicant pool	200+	99	1.2	24

others' findings that that 'diversity or excellence' is a false binary. The results also demonstrate that the active stance for diversity in hiring practices does not unduly exclude ORGs from being hired.

Variations on Scenario 1 As these are models, they facilitate considering variations to better understand the more impactful interventions at play. For example, Scenario 1 had several fairly significant EDI interventions. These included weighting the applicant pool with highly qualified Group B applicants, modelled by initially sampling their values from a uniform distribution between [50,100], whereas Group A values were sampled from a uniform distribution between [0,100]. The second intervention was sampling the Group A applicants themselves to reflect an application pool that had a 25% Group B representation. The third significant intervention was shortlisting the best two candidates from both groups. Of course there are regression analyses that will also provide similar understanding, but these models allow one to get a feel for the impact of the interventions directly. Table 2 illustrates the impact of modifying these interventions. Included in the figure is the statistics associated with the 30 by 30 objectives (30% new applicants for engineering registration in Canada to be women, by 2030).

The results summarized in Table 2 reinforce the observations made earlier in relation to the results summarized in Table 1. In addition, the results in Table 2 demonstrate how meeting a specific target (in this case, 30% representation) requires sustained effort even after the target. Results are not necessarily self-sustaining.

Tables 1 and 2 both provide comparative analyses. The simulation results are comparative in the sense that they may not provide the actual values of outputs but are useful to assess the comparative impacts of various strategies.

4 Discussion and Conclusion

There are a number of hiring best practices and policies that are emerging that are oriented towards inclusion or URGs. In Canada, some engineering schools are beginning to include representative diversity in the composition of faculty committees, mandate implicit bias training for academic search committees, set faculty-wide targets for URG representation on shortlists, conduct first-round blind interviews with a long list of candidates, have shortlists reviewed by the Dean for EDI criteria before candidates are invited to campus, include EDI-as-a-competency questions into the interview process, and take the time to extend or re-start a search if internal milestones are not met.

While these are excellent ideas at initial attempts to hire more inclusively, they are arguably more passive than the simulations undertaken here. If nothing else, these simulations illustrate the extremely long time constants in affecting change. This is likely exacerbated in a profession where the challenges are greater than in a university environment which in theory should be more open to a changing culture. Further, the simulation results also demonstrate the hollowness of a common argument that diversity incentives will sacrifice excellence in the professoriate.

Often, discussions related to EDI initiatives are considered to have exclusively a qualitative basis, and in data-driven professions like engineering, they are subtly discounted on this basis. These simulations highlight that the quantitative evidence for EDI is demonstrable and should be not only palatable but welcome in the engineering discourse. While any simulation has limitations, we argue that these insights are better than navel-gazing and provide a framework for proactively structuring change. As with all models, the assumptions and inputs used here can be varied to explore different initial conditions, sizes of applicant pools, target representations, and other variables. It is also worth noting that ABMs play a role here in providing a simulation that would be otherwise impractical or impossible to undertake in real life, as modelling and simulation are run for a simulated 50 or 200 year period.

It is axiomatically true that it is much easier to change the course early on than trying to steer a really massive ship later. Unfortunately for many engineering schools, we fall into the latter. Real structural change will require intestinal fortitude over a long haul.

References

1. M. Ayre, J. Mills, J. Gill, Yes I do belong': The women who stay in engineering. *Eng. Stud.* **5**(3), 216–232 (2013)
2. N.A. Fouad, W.-H. Chang, M. Wan, R. Singh, Women's reasons for leaving the engineering field. *Front. Psychol.*, 1–11 (2017)
3. N.A. Fouad, R. Singh, M. Fitzpatrick, J. Liu, *Stemming the Tide: Why Women Leave Engineering* (University of Wisconsin-Milwaukee, Milwaukee, 2012) Retrieved from http://www.daweg.com/documents/resources/Stemming_the_Tide.pdf
4. S. Ingram, Assessing the impact of career and family choices in mid-life: Striking the right balance for women engineers in their 40s. *Int. J. Eng. Educ.* **23**(5), 954–959 (2007)
5. J. Hunt, Why do women leave science and engineering? *Indust. Labor Relat. Rev.* **69**(1), 199–226 (2016)
6. Institution of Mechanical Engineers, *Stay or Go: The Experience of Female Engineers in Early Career* (Institution of Mechanical Engineers, 2017), available <https://www.imeche.org/policy-and-press/reports/detail/stay-or-go.-the-experience-of-female-engineers-in-early-career>
7. Engineers Canada, *2018 National Membership Information* (Engineers Canada, Ottawa, 2018)
8. Engineers Geoscientists Manitoba, *Annual Report 2016–2017* (Engineers Geoscientists Manitoba, Winnipeg, 2017)
9. S.A. Hewlett, M. Marshall, L. Sherbin, How diversity can drive innovation. *Harvard Bus Rev* (2013) available <https://hbr.org/2013/12/how-diversity-can-drive-innovation>
10. McKinsey & Company, 2018, Delivering through Diversity., available <https://www.mckinsey.com/business-functions/organization/our-insights/delivering-through-diversity?cid=other-eml-nsl-mip-mck-oth-1802>
11. World Economic Forum, 2019, The Business Case for Diversity in the Workplace is Now Overwhelming., available <https://www.weforum.org/agenda/2019/04/business-case-for-diversity-in-the-workplace/>
12. L. Smith Doerr, S. Alegria, T. Sacco, How diversity matters in the US science and engineering workforce: A critical review considering integration of teams, fields, and organizational contexts. *Engag. Sci. Technol. Soc.*, 139–153 (2017)
13. A.W. Woolley, C.F. Chabris, A. Pentland, N. Hashmi, T.W. Malone, Evidence for a collective intelligence factor in the performance of human groups. *Science* **330**, 686–688 (2010)
14. C.C. Perez, *Invisible Women: Exposing Data Bias in a World Designed for Men* (Random House, 2019)
15. E. Bonabeau, Agent-based modeling: Methods and techniques for simulating human systems. *Proc. Natl. Acad. Sci. U. S. A.* **99**(Suppl. 3), 7280–7287 (2002)
16. U. Wilensky, W. Rand, *An Introduction to Agent-Based Modeling: Modeling Natural, Social, and Engineered Complex Systems with NetLogo* (MIT Press, 2015)
17. W. Rand, R.T. Rust, Agent-based modeling in marketing: Guidelines for rigor. *Int. J. Res. Mark.* **28**(3), 181–193 (2011)

Developing a Scalable Platform and Analytics Dashboard for Manual Physical Therapy Practices Using Pressure Sensing Fabric



Tyler V. Rimaldi, Daniel R. Grossmann, and Donald R. Schwartz

1 Introduction

During either their junior or senior year, students participating in the Honors program are required to complete an Honors project and thesis. The goals of the project/thesis closely follow the overall goals of the Honors program, which appear on our website (Marist College):

Academic Excellence – To create enriched educational opportunities for capable, strongly prepared, highly motivated students that reflect a level of challenge suitable to this group both within and outside of the classroom.

Civic Learning and Leadership – To contribute to the development of the students' ability to lead, promote, and actively participate in civic learning projects that effect positive change, as well as their ability to be effective team members.

Integrity – To improve intellectual discourse among students and faculty through a meaningful curriculum of advanced coursework, seminars, and special events that demonstrate appropriate professional standards of behavior and respect for intellectual property and that demonstrate an active commitment to learning.

Global Citizenship – To enhance classroom learning with related experiences that encourage students to apply their special knowledge and skills in a way that serves others in the local and global community.

Continuous Learning – To offer eligible students challenging options for self-directed learning that may involve special projects within their majors or in the liberal arts core, and a culminating, distinctive work reflective of participation in the Honors program.

T. V. Rimaldi · D. R. Grossmann · D. R. Schwartz (✉)
Marist College, School of Computer Science and Mathematics, Poughkeepsie, NY, USA
e-mail: Tyler.Rimaldi1@marist.edu; Daniel.Grossmann1@marist.edu;
Donald.Schwartz@marist.edu

Students participating in our Honors program select their own projects and their own faculty mentors. This paper describes one such project.

The rate of professionals entering the field of physical therapy is expected to grow by 22% between 2018 and 2028. This predicted growth is greater than the predicted growth of any other healthcare-providing profession (US Bureau of Labor Statistics). The increasing number of physical therapy professionals brings an increase in physical therapy students; it is imperative that physical therapy education programs are adequately prepared to provide accurate and efficient teaching.

Currently there does not exist a tool for instructors to validate the correctness of their student's physical therapy movements. Quantitative metrics such as pressure, location, and consistency are not currently being tracked during physical therapy instruction. These metrics are crucial in teaching students precise movements and techniques and in evaluating the performance of students. This missing component has the potential to revamp physical therapy teaching pedagogies by providing useful feedback in real time. Students will no longer just watch their instructor's movements but actively following along, correctly matching their hand position and applied pressure on training devices. If a digital tool allowed physical therapy instructors and students to track those metrics in real-time feedback, then the quality of physical therapy education will have the potential to be enhanced, thereby producing better trained physical therapists.

Our analytics dashboard, in conjunction with pressure sensing fabric technologies, will fill in this gap by providing instructors and students the necessary quantitative metrics with an entirely digital and cloud-based solution. Instructors and students will be able to access quantitative metrics, data visualizations, and analytics, all in real time. The added convenience of cloud-based solutions introduces new potentials of remote teaching – a necessity for adapting to rapidly changing current events.

Key contributions of this paper include the following:

- Explores the motivation for a physical therapy analytics dashboard
- Describes the architecture of a physical therapy analytics dashboard
- Demonstrates new quantitative metrics in physical therapy
- Provides potential academic and professional adoption and implementations

The remainder of this paper is organized as follows: Section II discusses our prior work and introduces the definition of manual therapy and the concept of a physical therapy analytics dashboard. Section III describes our physical therapy analytics dashboard construction and development. Section IV concludes with a plan for future work.

2 Background

2.1 Physical Manual Therapy

Throughout this paper, we will use the term physical therapy to encompass the definition of manual therapy. As defined by the American Physical Therapy Association, physical therapists “are movement experts who optimize quality of life through prescribed exercise, hands-on-care, and patient education” (American Physical Therapy Association). There exist numerous different approaches to physical therapy, and of those many, we will address how our tool can benefit manual physical therapy – also known as manual therapy, or physical therapy as we shall call it throughout this paper. These therapies must be precise and accurate for successful results. Current training programs do not have pedagogical tools that reveal key metrics in real time such as pressure, location, and consistency. Our tool fills this gap and provides these metrics in real time.

2.2 Our Physical Therapy Analytics Dashboard

The current version of our analytics dashboard is a desktop application. Users will be able to download the dashboard application directly to their work machine. The supported operating systems are macOS, Windows, and Linux. This application is lightweight in nature and leverages our custom application programming interface (API) to efficiently make calls to our cloud-based back end. The front end (the downloaded application) features user interfaces that include real-time user authentication, sensor fabric port recognition (we call this our device driver), sensor fabric data collection, retrieval, and visualizations. In addition, we have included the foundation of what will become a major analytics suite. Our first offering is the feature for users to superimpose sensor fabric data graphs. Users can share their sensor fabric data with other registered users of the application. Our real-time superimposed graphs are the start to developing a new form of pedagogy that provides visualized quantitative metrics for physical therapy training. Instructors of physical therapy can leverage our pedagogical tool to store data from class sessions by using sensor fabric technology which can be shared to their students to begin practicing against their professors’ data sets. These metrics have never been able to be visualized and contrasted in real time, and it is exciting to pave the initial path of the future of physical therapy pedagogical tools.

3 Physical Therapy Analytics Dashboard Construction

3.1 *Studio 1 Labs Sensor Fabric*

It is important to understand the hardware that we are using in conjunction with our software. Studio 1 Labs, “a technology company focused on evolving everyday objects with fabric sensing technology” (Studio 1 Labs), provided us with a robust pressure sensing fabric (see Fig. 1). Their pressure sensing fabric is highly adaptable and can be used on irregular surfaces, crucial for a project that focuses on analyzing inconsistent areas. The sensor fabric is densely populated with 64 sensor sensors, in an 8×8 matrix. The dense number of sensors allows for precision and accuracy. Our current system leverages its direct connection capabilities; we are able to plug in the sensor fabric to our machine and let it begin its reading without any added configurations (as you will see our application communicates with the device automatically, handling this for the user). Additionally, the fabric can be used wirelessly via Bluetooth connectivity if a user’s device allows for such communication.

Our system is designed to be used by faculty and students within the Marist College Doctor of Physical Therapy Program. Faculty members will demonstrate various physical therapy maneuvers, placing the sensor fabric on the client and applying pressure as appropriate. Our system will measure and collect the sensor readings, storing them for future evaluation. Students will then use the fabric as they attempt to duplicate the maneuvers. Our system will collect their readings and then show the results of comparing the professor’s readings with the student’s readings. Our system can also be set up so that the professor’s recorded readings and the student’s current readings are shown “live” on the screen, so that students can adjust their pressure and position in real time to more closely match those of the professor. (See Fig. 2.)

Fig. 1 Sensor fabric sheet provided by Studio 1 Labs. This fabric is Bluetooth enabled and is also capable of establishing wired connections to stream data directly to our application



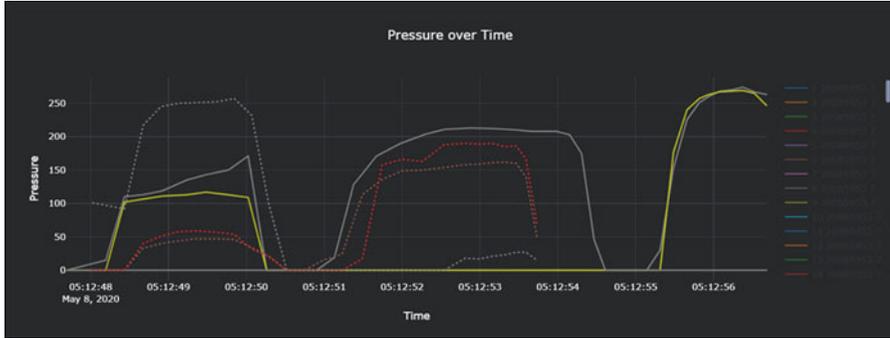
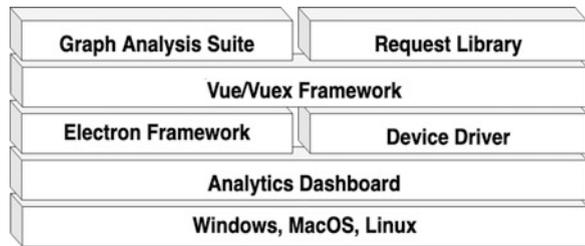


Fig. 2 Sample overlay comparing professor’s data to student’s data

Fig. 3 The environment that hosts the front end of our application. Every container mounted on top the “Analytics Dashboard” container is what makes up that component



3.2 Dashboard Development: Front End

Our front end consists of many different and related technologies including ElectronJS, VueJS, Bootstrap, and Plot.ly. (See Fig. 3.) Each of these technologies makes up our robust dynamic user interfaces. To start breaking down how we used each technology, we will start with the foundation: ElectronJS and VueJS.

ElectronJS is a framework that we used to build our desktop graphical user interfaces. It combines the Chromium rendering engine and Node.js, a JavaScript runtime environment. Electron is composed of multiple processes: renderer and browser. The renderer process loads HTML and CSS views that appear on the user’s screen, and the browser process contains the application logic. On top of ElectronJS we used Vue, an interface framework. Vue provides us an object-oriented-like approach to building user interfaces: allowing us to reuse components and seamlessly pass data through the interfaces. Vuex, a state management pattern library for VueJS, served as our centralized store for all components in the ElectronJS application, providing rules to ensure “that the state [of a component] can only be mutated in a predictable fashion” (Vuex). With our base to our desktop application, we can start mounting other technologies such as Bootstrap and Plot.ly.

When creating user interfaces, we chose to use Bootstrap, a framework providing adaptable and responsive CSS styles. The Bootstrap framework has preset interface components which we molded into our application. We scaled each used Bootstrap

Session	CWID	Description
1	80085953	Joint traction/distraction sample movement 1
2	80085953	Joint traction/distraction sample movement 2 with instruction
3	80085953	Caudal glides demonstration

Fig. 4 This table shows the recorded sessions. A user can graph a sensor session by selecting a session by its session number. They will then have the option to graph the session as a line graph (Fig. 5) or a heat map (Fig. 6)

component by encapsulating them into VueJS component templates for multiuse around the entire application. This property of “multiuse” significantly enhanced our development experience as it greatly reduced redundant code. In addition, it provided us with a common interface for common components used in the application. After applying our necessary styles and backbone to the application, we implemented [Plot.ly](#), a data visualization tool built on top of D3JS, for our data graphics. We specially implemented line graphs and heat maps, as we believe they allow us to visualize sensor data most clearly.

For data to be streamed into our application, a user must connect the Studio 1 Labs’ pressure sensing fabric to their machine via wire or Bluetooth. Our application will quickly recognize the port that the fabric is using and establish a real-time data pipeline to our application. Because Vuex provides state management, our application dynamically visualizes pressure sensor data. The user has the option to save their pressure data (with metadata such a description) or to discard it and restart the data stream. The user also has the ability to query their data by interacting with our request handler. Our request handler will then query our back-end API to then return data to a table. The user can then select which data to visualize (Fig. 4) and can specify which settings to display, such as a line graph (Fig. 5) or a heat map (Fig. 6), and whether to superimpose the user’s data on top of the instructor’s data, so that a comparison between the pressure the student is using and the instructor’s pressure can be shown.

The coordination of these technologies allowed us to develop scalable interfaces in a web application development environment. Since we were developing in a web application environment, our desktop application can be ported to a web-accessible application very easily. In fact, the two applications (the desktop application and web-accessible application) could perform the same functions and provide the user a similar experience. The user can then specify whether they prefer the look and feel of a browser application or a desktop application.

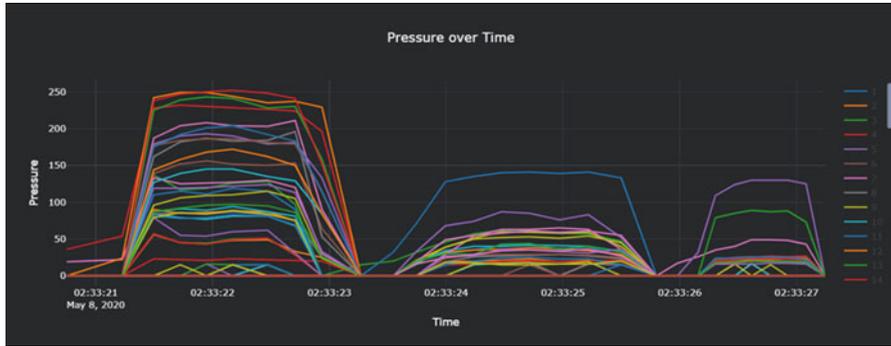


Fig. 5 Line graph visualization of a session via individual sensor outputs

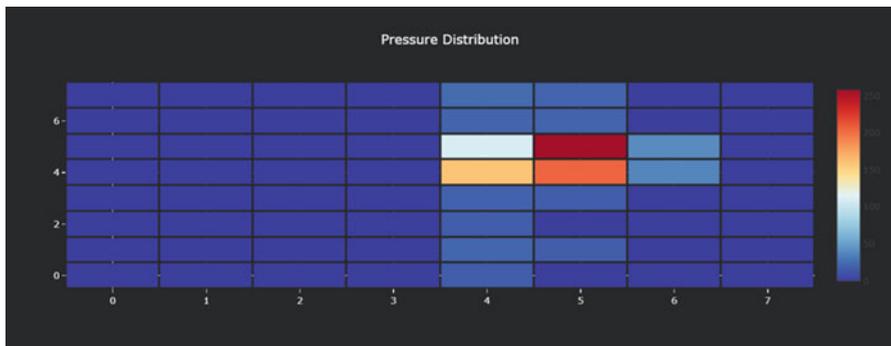


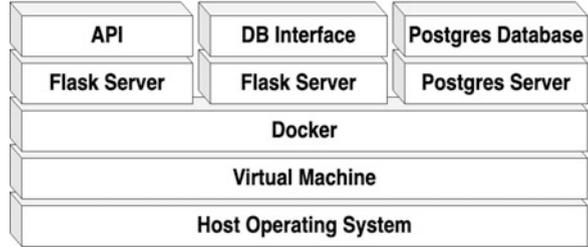
Fig. 6 Heat map visualization of a session targeting a specific region of the sensor fabric

3.3 Dashboard Development: Back End

A **microservice** (in our architecture) is some process that communicates over a network using HTTP. Microservices perform specific tasks and are entirely independent from other microservices. Our microservices communicate over HTTP and are based on the **RE**presentational State Transfer (REST) protocol. Each of our microservices performs crucial tasks that are independent of the other microservices. As seen in Fig. 7, our architecture features three microservices: our application programming interface (API), our database interface, and our database.

Our API was developed using Flask, a lightweight micro web framework written in Python (Flask). The API simply acts as an authenticated interface to the front end and as an authenticated gateway to our microservices’ interfaces. Our API controls the flow and format of data by making efficient use of Python to perform any data manipulations before reaching our front end. This speeds up transactions dramatically as our front end is based on Node.js, which is single threaded by nature. Our database maintains authentication for interacting with other services (each of

Fig. 7 Microservice architecture diagram presenting our virtual environment that hosts the back end of our application



the other services also include authentication but are validated through our API). This provides an added level of protection across all of our microservices.

The database interface was also developed using Flask. It provides definitions, models, and migrations necessary to interact with our database. This interface provides a channel to our API that enables it to communicate to our database. The API does not need to have knowledge of the implementation of the database; instead, it only needs to know about the HTTP routes provided by the database interface. In this sense, we have microservices requesting information from other microservices. This can be seen in miniature applications interacting with other miniature applications.

Our last microservice is our PostgreSQL relational database. We chose to work with relational data over a non-relational database after we realized the data collected would eventually develop relationships between the actors (students and professors) participating in our application. In its current state, the complexity issue arises when we relate professor sensor data to student sensor data and vice versa. This seemed like a great opportunity to utilize a relational database to preserve data integrity. To communicate with the database, requests must enter through the database interface which contains the logic of accessing data and maintains authentication.

Our back-end architecture is solely based on microservices. The reason for this is to address the following concern: what if a new developer joins the team and is tasked with adding a new feature (such as an analytics suite) to the back end? By leveraging our microservice architecture, the new developer would only need to use our API documentation to interface with any of the existing microservices. The developer would not need to understand underlying details for each microservice – instead, they can simply use the API, which naturally orchestrates communication across the services that are involved with that specific request. The developer's focus could be shifted entirely to developing the new feature and only that feature. Once the feature has been developed, it could be treated as a small application. As such, the developer would simply need to update the API and point HTTP routes to their microservice for use.

3.4 Dashboard Development: Deployment

Our back-end component of the application is deployed to a cloud platform (in our case a virtual machine hosted in the Marist College Enterprise Computing Research Laboratory) seamlessly via Docker containerization. Docker allows us to package our back-end components into standalone, dedicated Unix environments (Ubuntu in our implementation), also referred to as Docker containers. This allows us to be modular in nature and scale our microservices for fault tolerance. In addition, this also allows us to isolate each container. This isolation adds a layer of security to the stack as containers exist independently from each other – unless explicitly told otherwise via Docker subnet, as we have done.

As depicted in Fig. 7, three containers are mounted on top of the Docker virtual machine. Our deployment time was drastically reduced when we switched to Docker, as it simplifies the development environment requirements needed on the cloud platform used for deployment; each container provides the necessary details and components needed to run. In essence, the host machine is only required to have Docker installed. Once installed, it is as simple as grabbing our Docker image (the container before it is deployed) and running an instance of it on the Docker-ready machine. Everything else is “magically” handled through our development of the back end. This easy-to-deploy aspect is crucial in making our application adoptable and implementable across different institutions.

4 Conclusions and Future Work

It is evident that physical therapy instructors need a new pedagogical tool to enhance the precision and accuracy of their training. Our physical therapy dashboard, in conjunction with pressure sensing fabric technologies, can help to fill this role by providing instructors and students with ways to capture and present quantitative metrics such as pressure, location, and consistency. Our dashboard provides an added level of convenience by leveraging our custom cloud-based solutions. This convenience provides instructors and students with real-time access to quantitative metrics, data visualizations, and analytics. Our dashboard can be scaled to meet the institution’s demands and can be deployed to any Docker-enabled system, allowing it to be institutionally agnostic and rebranded accordingly.

Our future work entails system testing and additions to our analytics suite. We plan to continue our collaboration with Studio 1 Labs and the Marist College Doctor of Physical Therapy Program. We have started preparing an institutional review board (IRB)-approved study to assess the educational impact this tool will have on academic physical therapy professors and their students. We plan to conduct testing in one or more classes during the upcoming Fall semester.

Sources

1. Marist College (2020) Honors Program. <https://www.marist.edu/honors/student-learning-outcomes>. Accessed 18 Jun 2020
2. American Physical Therapy Association (2020) Becoming a Physical Therapist. <https://www.apta.org/your-career/careers-in-physical-therapy/becoming-a-pt>. Accessed 18 Jun 2020
3. US Bureau of Labor Statistics (2020) Physical Therapists. <https://www.bls.gov/ooh/healthcare/physical-therapists.htm>. Accessed 18 Jun 2020
4. Electron (2020). <https://www.electronjs.org/>. Accessed 18 Jun 2020
5. Node (2020). <https://nodejs.org/en/>. Accessed 18 Jun 2020
6. Flask (2020). <https://flask.palletsprojects.com/en/1.1.x/>. Accessed 18 Jun 2020
7. Docker (2020). <https://www.docker.com/>. Accessed 18 Jun 2020
8. Vue (2020). <https://vuejs.org/>. Accessed 18 Jun 2020
9. Vuex (2020). <https://vuex.vuejs.org/>. Accessed 18 Jun 2020
10. Plotly (2020). <https://plotly.com/>. Accessed 18 Jun 2020
11. Bootstrap (2020). <https://getbootstrap.com/>. Accessed Jun 2020
12. Studio 1 Labs (2020). <https://www.studio1labs.com/>. Accessed Jun 2020

Tracking Changing Perceptions of Students Through a Cyber Ethics Course on Artificial Intelligence



Zeenath Reza Khan , Swathi Venugopal, and Farhad Oroumchian

1 Introduction

Mankind is an intelligent species whose individual and communal progress has largely been an interdependent effort [27]. Over the years, this interdependence has been strengthened and elevated with technological advances. Technology's bounty has been in steady progress evident from the power of smart phones in our hands. The fourth industrial revolution has brought new dimensions to the evolution of technology, with biological, digital, and physical spheres, pushing open new frontiers in many industries with new technologies such as block chain, 3D printing, internet of things, and so much more. Particularly, artificial intelligence "has been described as the fourth industrial revolution" itself [8]. This evolution has been triggered by the advancement of machines that are now able to intuitively developing knowledge which goes beyond simple if-then steps and has opened new arena that are pushing boundaries of jobs, economies, customer satisfaction, predictions, medical and climate simulations, and so on.

Artificial intelligence (AI) has become an exciting area of technological development with the promise of changing the world as we know it. Although majority of the population's understanding of AI comes mostly from movies, some of which almost always is negative, Bill Gates has stated that it is "promising and dangerous" [7].

With the UN Sustainable Development Goals focusing on quality education, building resilient infrastructure, promoting sustainable industries, reducing poverty

Z. R. Khan (✉) · F. Oroumchian
University of Wollongong in Dubai, Dubai, UAE
e-mail: zeenathkhan@uowdubai.ac.ae

S. Venugopal
Love That Design, Dubai, UAE

and hunger, and encouraging gender equality, good health, responsible consumption and more, artificial intelligence is playing a crucial role in shaping industries and lives in ways not imagined or comprehended before [33]. From increasing productivity to providing green solutions, smart living to increase accessibility and inclusivity, automation and artificial intelligence has the potential to do tremendous good for the world [16]. However, it has the potential to have negative impact as well, enlarging already existing digital divide, increasing carbon footprint, and so on that can backfire on achieving the same goals that AI can help to achieve, depending on geographical, political, and economic standing [35]. More specifically, in a report by top AI experts, “speech synthesis for impersonation,” “analysis of human behaviors, moods and beliefs for manipulation,” use of “thousand micro-drones” as weapons, and attacks using and controlling autonomous cars “causing them to crash” are just some other possible arenas of use of AI that make the future threatening [6].

For this reason, in order to prepare future leaders, policy makers, programmers, and developers to be able to develop and use AI in a manner that benefits the greater community and helps achieve the United Nations Sustainable Goals and by extension UN 2030 Goals, it is imperative to not only teach students about the technology and prepare them to develop future systems but also to understand the implications, repercussions, and responsibility with which such technology should be developed and used.

Hence, studies on ethical use of artificial intelligence is emerging as more educators identify the need to infuse values in the current generation to protect future digital leaders [29], premise for this study that aims to record student perceptions of artificial intelligence and automation during a capstone lecture for an IT ethics course at a tertiary education institution over a 10 year period in order to understand how, if at all, students view artificial intelligence, how their perceptions may have changed and how they are influenced by the ethical discussions partaken during the lesson.

2 So Why Cyber Ethics?

Every year there are numerous small and large cases of cyber-attacks in every country. In 2018 alone, identity thefts affected about 60 million people in the United States [31]. This led to committing \$15 billion for cyber security in the fiscal year of 2019 which is still a partial amount of entire cyber budget [31]. In 2018, a whistleblower called Jack Poulson, a former Google employee, created a non-profit initiative to question Google’s plan about building a censorship AI for the search market in China. Although the fundamental purpose of the project was to build more natural search sets for the users and re-enter the Chinese market, it clearly was a customized Google search engine tailored as per the Chinese government’s needs of surveillance and censorship for which Google’s very own Sundar Pichai had told his employees to argue that this was an “exploratory project” [11]. If this is the case

for the world's best search engine, the world's famous social media platform is no exception to cyber breaches and controversies. Without proper consent, Facebook sold the data of 50 million users (which was later verified to be 90 million) to a political consultancy called the Cambridge Analytica and third parties ahead of the then US Presidential Campaigns in 2016 [30]. The data collection started in the form of a survey collected by a Cambridge University researcher Aleksander Kogan who paid a small fee for participating in the survey. When survey app was installed by the survey taker, it shared the information of the firsthand users of the app and their friends without consent. This Cambridge Analytica scandal is a significant one in cyber history that highlights the plight of naive digital users who assume that Facebook is "safe" for all. As dreadful as it gets, the damage done by the data malpractice of Zuckerberg does not stop at the Presidential elections. To date, nobody knows how the dataset sold has been used by all the companies that purchased it and is suspected to have fallen beyond borders – possibly into foreign hands [30]. Later in the year, Wall Street Journal reported that Facebook also accepted to have been hacked by non-political "criminal spammers" who gained access to sensitive data of 29 million users and their linked accounts [18]. This insensitive and irresponsible attitude by Facebook acquired them the title "Digital Gangsters" from the UK Parliament accusing them of an unethical attitude and performing thoughtless activities "intentionally and knowingly" [24]. Sadly, it is these corporations or conglomerates that currently run the tech show for the entire world and our future iGens are growing along with them.

3 Cyber Ethics for iGens

iGens are those children typically born after 1995 and grew up along with the internet – a generation that does not know a world without it. This generation is also called Generation Z as they succeed the Generation Y – who are also tech-savvy but not as good as the iGens [19]. However, this tech savvy generation does more than just share the information online – it produces, shares, and governs the information available over cyberspace. The cyberspace is the field where all the information is stored and shared electronically. Therefore, ethical issues are the result of this unwarranted and unverified sharing. Advancement in the usage and sharing of information are the primary reasons why cyber ethics should be considered an important aspect in today's education. While traditional ethics are based on beliefs, norms, and goals that can differ as per various ethical principles, cyber ethics are those that show the responsible use of cyber space by governing its procedures, values, and practices [32]. Since the internet age is relevantly young in ethical experience, users of the internet need to be taught how to ethically use the same. Often, digital natives find themselves in an ethical dilemma in spite of underlining the proper use of Information and Communication Technology (ICT). Robert Kruger highlighted that 8 out of 10 students in the United States with internet access in school download unauthorized and unlawful material simply

because they are “unaware” that it is wrong [14]. Most of this happens without understanding the serious consequences behind such acts. If students are taught the ethics behind shoplifting alongside its consequences, he/she would think twice before acting upon a thought. However, digital natives are only sprinkled with technical terms such as software piracy, plagiarism, and digital property rights without categorizing them as digital “stealing.” What happens if these seemingly “unaware” generation grows up blind to numerous attacks and security breaches that happen worldwide each day? [14]. So the fact remains – education is the foundation for individual progress and a crucial aspect in the contribution toward the society. Studies have shown that education is the key to bridging the gap between classroom and workplace [13]. Cyber-attacks can spark from anywhere for many reasons – it can be a corrupt employee, unhealthy business competition, criminal groups, or political conspiracy. However, only technical measures may not be the solution to effective defense. A good cyber defense strategy includes appropriate education about risks and awareness of the cyber world [27]. This gives rise to the need for education in technological advancements for which their associated moral education is imperative. To make proper sense of the different options available over the internet, parents and children are recommended to classify and conceptualize the options for real examples to determine the right from the wrong. Parents and children must not assume safety due to upcoming filtering systems and firewalls – rather education of limits and dos and don’ts alongside their consequences are known to work better. Illustrating the actual harm that could be done by plagiarizing or spreading online hate can reduce malicious behavior that wastes people’s energy and time [34]. For this reason, Sobiesk et al. proposed a mandatory multi-disciplinary approach to netizens to get a broader understanding on the emerging cyber crisis and opportunities. This era of a digitally divided society demands the facilitation of cyberspace that is currently clouded with ethical issues that are “socio-cultural and academic” in nature [21].

4 Artificial Intelligence and Cyber Ethics

Artificial intelligence is, undoubtedly, man’s most intelligent invention. Artificially intelligent systems are programmed to follow a series of algorithms. That is why some consider AI systems to be unbiased or fair or ethical or emotionally neutral. Yet, the devices that display intelligence are programmed by intelligent humans who may choose to follow an unethical course or machine who might learn our flawed and biased behavior by observing and learning our behavior. That is in the heart of research and development for programming moral agency into new technologies to manage real life situations and make decisions ethically and fairly [1]. Frameworks that develop autonomous reasoning and thinking inherit the values of the human society based on sociocultural norms. Thus, the field of AI involves weightier matters for consideration such as responsible reasoning, data stewardship, and transparency [9]. Apart from considering the practical function

of any technology, the ethical functioning of the same is crucial in attaining fair and reasonable state of affairs. For example, an ATM machine must be ethically tuned to count the right amount of bills to ensure fair transactions. Implicit ethics stand at the core of computer software development, absence of which would rule out the usage of computers altogether. However, an AI agent might require more explicit representation of ethical decision-making processes. A computerized game of chess can be example of explicit ethics that can judge and calculate the next move on the board [20]. Even though explicit ethics on machines sounds elusive today, our debates are moving toward whether we should develop lethal autonomous weapons [27].

According to Vinuesa et al. [33], artificial intelligence can play a significant role in achieving the United Nation's Sustainable Goals (UN SDG). In fact, the study posits that AI can "enable the accomplishment of 134 targets across all the goals, but it may also inhibit 59 targets" [33]. From "using satellite imagery to start to track how reforestation is progressing; to track(ing) livestock as a means of predicting conflict . . . [to tracking] smarter traffic signals to reduce congestion and pollution", McConaghy [16] has recorded ways AI can help achieve the goals.

This duality of the technology then makes it crucial that we ensure we are preparing the next generation of tech-savvy graduates who join the workforce are in fact versed in the ethics and moral principles that should govern the development and use of artificial intelligence.

Therefore, this study's objective is to *attempt to capture if student understanding of artificial intelligence and its ethical impact has changed over the years and whether teaching about it as part of an ethics course has any real impact on how they perceive artificial intelligence in today's world.*

5 Methodology

In order to conduct this project, one lecturer's experience in teaching a cyber ethics subject at an offshore campus of a Western university has been captured. The subject has been a part of the core degree both for business and IT, then was moved to becoming a core for only engineering and IT, and then as an elective for others. Currently, the subject is a final year elective for IT and engineering students.

Students taking the subject range in age from about 18 years to 22 years. The range is attributed to fact that the subject was originally a second-year subject, with students toward the younger spectrum till 2014/15 and moving to the higher end of the spectrum with more almost-graduating students taking the course.

Demographically, students were from various ethnic, cultural, and curricular background, enrolling in the subject.

The subject included wide and diverse topics to cater to the learning objectives as follows:

- Identify the privacy, legal, and security issues related to the introduction of information and communication technologies
- Explain solutions to security and privacy problems arising from the introduction of technology
- Evaluate the impact of information technologies through the application of ethical frameworks
- Explain the role of professional ethics codes of conduct
- Demonstrate the understanding of the need for social computing and ethics in cyber space

The subject typically has three to four case deliverables, one group research project, an individual reflective essay, and reflective journaling. Key focus for this study is the class discussions during lectures and the reflective journal posts.

6 Sample Size

Students were always encouraged to participate in the journaling experience which also was indicative of class attendance and participation in discussions during lecture sessions. However, a retention rate of response and engagement was approximately 75%. Ohme et al. [22] have posited that the accepted rate of response in a study is about 66%, and while this refers to research studies, considering the student engagement in posting journal entry was counted is therefore drawn on as response that is studied for this project. In this respect, 75% is a good rate of response. Given this understanding, total number of responses recorded was $n = 1503$ from years 2009 to 2019 (Table 1).

6.1 Setting Up the Journal

Self-Reflective Journaling (SRJ) was introduced first as written submission and later (from 2014) as online using learning management system; after each lecture, students were expected to post their thoughts on the concepts covered and comment on the discussions that took place in class. This primarily worked as a revision and increased class attendance. SRJ helped record what students thought of the topic discussed in class. It also helped those students who were shy or introverted to voice their thoughts and put forward their views. Students had the option to share their view either with everyone in class or only with lecturer. Students independently

Table 1 Distribution of responses across study period 2009–2018

2009	2010	2011	2012	2013	2014	2015	2016	2017	2018	Total
212	196	240	211	198	80	78	98	92	98	1503

researched the topic, uploaded videos, brought in articles for discussion with others, and so on. Overall, it encouraged active student participation, critical thought and application, and reflective writing [12].

SRJ was set up to track student understanding of ethics and how it grew or changed over the semester. Students were expected to post at least one entry every week starting Week 4 after every lecture. Although toward the beginning the students' entries mostly included a summary of the lecture or how good the lecture was, they soon understood that wasn't adding value. They began to think about the topic and then started writing posts that critically analyzed the topic [13].

6.2 *Lecture on AI*

The lecture on artificial intelligence was considered the capstone lecture for the subject and took place on the last lecture of semester. By this time, students had already attended lectures on topics such as networking, social media, computer reliability, privacy, security, intellectual property, and professional codes of conduct. Students also had lessons on ethical theories such as Kantianism, Utilitarianism, Locke's Theory of Property, Social Contract, and others to provide frameworks that students can relate to for decision-making [23].

The teaching revolves around creating "times for telling" [28] by first introducing the topics with guided discovery activities where the students work in collaborative groups, progressively arguing their ideas and opinions by collaborating around their formal and informal knowledge and experiences. The lecturer then creates an "analogous relationship" [10] between stakeholders through scenarios, videos, and questions and gets students to discuss the possible answers. The lecturer then draws on "both sides toward the middle" technique, involving students in extremes of a continuum of an ethical dilemma. Because all students can see that each end is an extreme situation, they get interested to find out what they and their peers would think and how they would solve the dilemma toward the gray areas. Lecturer "tests the limits" [10] of the students' understandings of concepts by changing the facts of a case until the decisions begin to differ, allowing students to "see" the influences and reasoning behind their own decisions. Lastly, the lecturer applies the "writing and policy and reversing... role" where students engage in writing out actual policies they would want to implement based on the case discussed and then are asked in a Kantian fashion [17] to reverse their role to see if they could live under such a rule.

By the last lecture, students are used to this model of teaching and when artificial intelligence theories and cases are introduced, they get into the discussion, often splitting into four segments as shown in the figure below (Fig. 1).

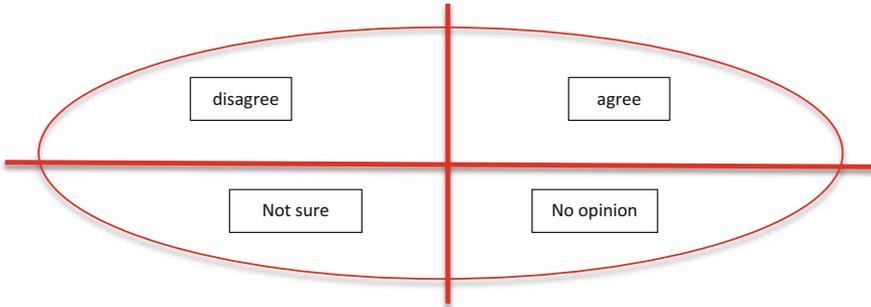


Fig. 1 Class discussion segments (CDS)

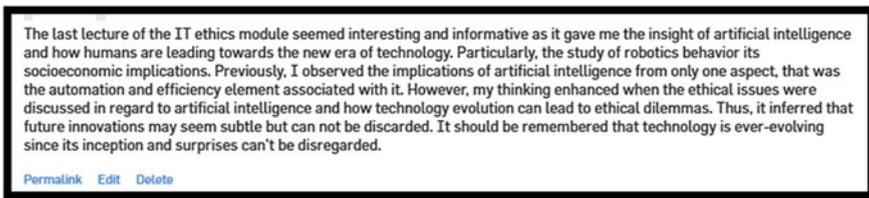


Fig. 2 Sample post 1

7 Results

Entries were analyzed using qualitative coding. Of interest for this study were the posts on the last capstone chapter which focused on Artificial Intelligence and Robotics. Below are some sample posts:

7.1 Sample Posts (Figs. 2, 3, 4, 5, and 6)

Students’ attitude during lectures was captured based on where they stood during the CDS activity in class for questions such as “do you think artificially intelligent robots should have human characteristics”, “do you think AIR can or should be AMA (artificial moral agent),” “do you think AIR should be weaponized,” “do you think AIR should have capacity to decide on collision decisions,” and so on (Fig. 7).

Using the circular segmenting of lecture discussions shown in Fig. 1, the answers were distributed across the years as illustrated in Table 2 and Fig. 8.

Another dimension of result was based on class discussions from capstone lecture session and the journal posts, and these results were quite fascinating.

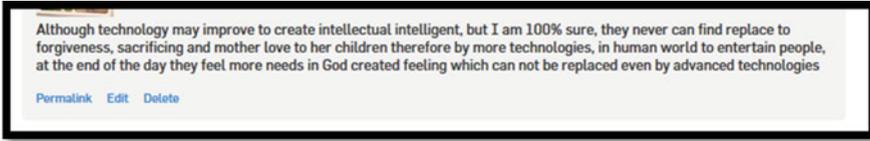


Fig. 3 Sample post 2

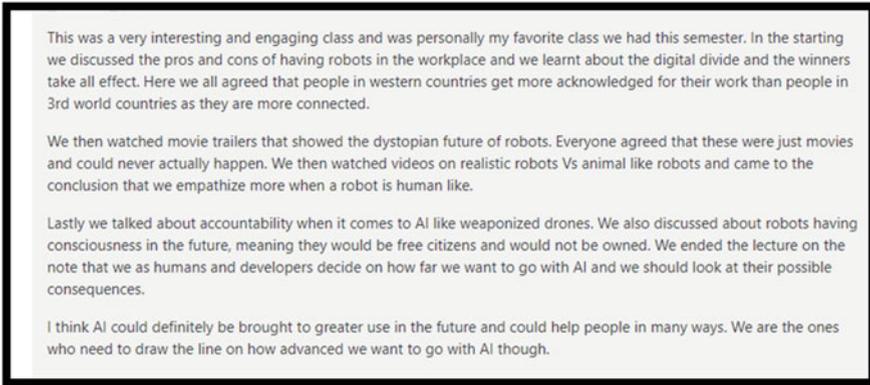


Fig. 4 Sample post 3

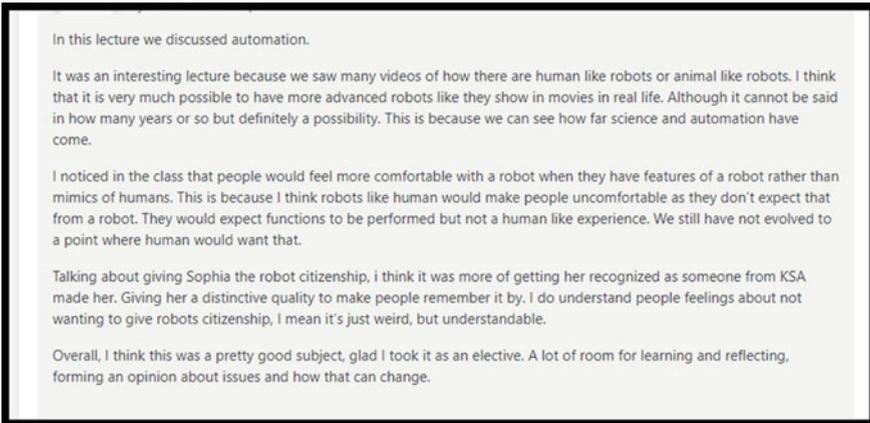


Fig. 5 Sample post 4

Toward the beginning, more students seemed to think that the concept of artificial intelligence and automation was “far-fetched,” “far from reality,” and “to be used with responsibility.” This is reflected in their choice of segment during the class activity where majority seemed to disagree with the questions posed on immersion of artificial intelligence and moral decisions.

this week's class was about robots and how they might change our future.

in the last 10 years ago so many things have been automated that have changed the industries in good and bad ways. automation could decrease the costs, reduce any errors that humans might make during any given task, produce the goods in much quicker time and the quality of the production is much higher. These are the good things about automation. but alongside these also humans are required to have higher skills in order to find jobs because basic tasks could eb covered but robots and there will be less jobs for humans as well.

there was a video that showed a dog alike robot and how stable it was. most people found it not very interesting but, in my opinion, it was a very advanced robot that had so many amazing features. it had an amazing balance and stability and could adjust to walking on different surfaces with different obstacles on its way. even humans that have eyes and can see and analyze the situations, they might fall down sometimes. but there was a scene in the video that was showing the robot, it showed someone kicking the robot without the robot seeing it and the robot could easily stabilize itself without falling down. if the same thing had occurred to humans or any other animals, there was a high chance that they would fall on the ground.

then there was another discussion about human robots made in japan. their skin was so human alike and also their movements were almost natural. there was a very interesting point in the video, the person was speaking about the robots when they were sent to hospitals for testing, he mentioned in the meanwhile that these robots were gone, the crew had missed them and felt absence. that means that even though they are not living creatures and completely human made (they have no brain of their own and every single detail of them is programmed and pre-defined by humans) humans can still interact with them and get attached to them like they would to a normal person.

Fig. 6 Sample post 5

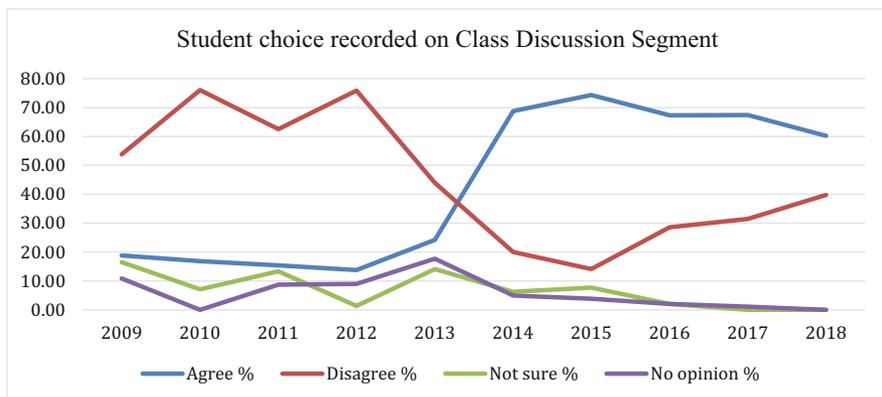


Fig. 7 Student choice on CDS

In later years, more students found it “fun,” “very real,” “already here,” “cool,” “no big deal,” “great progress,” “quick and effective in different areas,” “reduces human error,” and “can be weaponized.” This is also reflected in the choice of segment where more students significantly stood on “agree” – that is, resulted in more positive than negative.

What is more is, using Bletzer [3]’s method of visualizing qualitative data to create Word Clouds, feedback from students were segmented from 2009 to 2013 and then from 2014 to 2018. Word Cloud is “a visual representation of a collection

Table 2 Response rate of student decisions in percentage

Year	Agree (%)	Disagree (%)	Not sure (%)	No opinion (%)
2009	18.87	53.77	16.51	10.85
2010	16.84	76.02	7.14	0.00
2011	15.42	62.50	13.33	8.75
2012	13.74	75.83	1.42	9.00
2013	24.24	43.94	14.14	17.68
2014	76.39	22.22	1.39	0.00
2015	74.36	14.10	7.69	3.85
2016	67.35	28.57	2.04	2.04
2017	67.39	31.52	0.00	1.09
2018	60.20	39.80	0.00	0.00



Fig. 8 Word count for most used words or strings of words in online journals from 2009 to 2013 and from 2014 to 2018

of text documents that uses various font sizes, colors, and spaces to arrange and depict significant words” [5].

As the study’s purpose was to capture student feedback, the qualitative data were extracted from the journal entries for the topic of AI and Automation. The data were then organized in alphabetical order. Many students used similar terms, so this exercise helped identify the top 20 words or strings of words to describe how they felt about automation and ethics. Based on the frequency and emphasis by students, the words were then arranged in order from 20 down to 1 (see Table 3). An online free builder, WordClouds.com, was used by uploading the extracted and cleaned data .csv file. Maramba et al. [15] posited that using web-based text processing tools such as word cloud sites was useful in providing “quick evaluation of unstructured . . . feedback” and such tools have been used to generate the word clouds shown in Figure 8.

Table 3 CSV list of words or strings of words

List A – 2009–2013	List B – 2014–2018
weight;“word”;“color”;“url”	weight;“word”;“color”;“url”
17;“cannot be trusted”;“”;“”	18;“carefully observed”;“”;“”
16;“good for new jobs”;“”;“”	18;“cautious”;“”;“”
15;“cause job loss in factories”;“”;“”	17;“no citizenship”;“”;“”
14;“should be cautious”;“”;“”	16;“no human skin”;“”;“”
13;“huamn decision cannot be trumped”;“”;“”	15;“should have citizenship for accountability”;“”;“”
12;“cannot have human characters”;“”;“”	14;“depends on how we use it”;“”;“”
11;“far-fetched”;“”;“”	13;“eases work for humans”;“”;“”
10;“far from reality”;“”;“”	11;“feel comfortable”;“”;“”
9;“cannot be moral agent”;“”;“”	10;“fast”;“”;“”
8;“cannot replace God created feeling”;“”;“”	9;“can be allowed to make decisions”;“”;“”
7;“ever evolving”;“”;“”	9;“endearing”;“”;“”
6;“efficiency”;“”;“”	8;“emphatize more when robot is human like”;“”;“”
5;“entertainment value”;“”;“”	7;“efficient”;“”;“”
4;“cannot go too far”;“”;“”	6;“no big deal”;“”;“”
3;“nothing to worry about”;“”;“”	5;“precise”;“”;“”
2;“no replacement to forgiveness”;“”;“”	4;“very real”;“”;“”
1;“interesting”;“”;“”	3;“fun”;“”;“”
1;“no replacement to sacrifice”;“”;“”	2;“great progress”;“”;“”
1;“automation”;“”;“”	1;“innovation is a must”;“”;“”
1;“can be an ethical issue”;“”;“”	1;“it is the future”;“”;“”

8 Discussion and Lessons Learned

The results of the findings are very interesting when looking at how the generations of students’ choices have changed over the years toward artificial intelligence and artificially intelligent beings. Three significant results are noteworthy:

1. Over the years, students’ perception of artificial intelligence has changed to become significantly positive.
2. Students do not fully comprehend all the implications of artificial intelligence.
3. Irrespective of how they view the technological advancement, actively engaging in discussions on the moral principles and ethical dilemmas help students discuss and debate the necessity for transparency and predictability in decision-making based on both the development and use of artificial intelligence.

When looking at students’ attitude through their discussion in class and journal posts, one obvious reason for the difference between 2009–2013 and 2014 and after may be the speed at which technology in general has advanced in the last decade, and specifically AI has immersed into daily lives in the form of Siri, Alexa, and so on. Millennials grew up as technology and information took over. In contrast, iGens

were born into technology, growing up depending on technological devices and decision-making daily [2]. Moreover, it is important to note here that the significant change that takes over the years could also be attributed to millennials graduating and iGens coming into classrooms. While millennials were and remain slightly skeptical of artificial intelligence and slow to adopt, iGens are more fluent, easily adopting to changing technological advances, finding the features and advancement as fascinating and used to interacting with artificially intelligent devices and apps [35].

It is also important to observe that while students in the early phase of the study were more skeptical of the AI and its potential and readily agreed that caution and responsibility were required, the students in the latter phase were enthusiastic about the technology, but when discussion ensued, they were more intrigued and wanted deeper understanding and discussion on issues such as “citizenship of machines such as Sophia” and “not having human-like features or skin because that made them feel uncomfortable,” particularly as they engaged in ethical debate in class and then posted their journal entries online, highlighting impact of the ethics class [12].

These are important findings, giving educators and researchers an understanding of how fast and varied the generations’ views of AI advances are, how little they understand in terms of the real possibilities of AI, and how they may be shaped.

Importance and effectiveness of teaching ethics in AI to students can also be shown through the findings of this study. Upon following discussions and engaging in activities in the lecture, students, whether millennials or iGens, do seem to respond to the theories and concepts, nudging them to look at all perspectives, not just accept AI but question our dependencies on them to decide where and how to be responsible with decision-making.

9 Conclusion

This study focuses on AI and rise of AI advancements with the advent of the fourth industrial revolution. The primary reason for focusing on AI is its potential to benefit the society, economy, environment, the rate at which AI is becoming popular, and the constant demand and need to create more “thinking machines” [4]. Introducing future developers, innovators, and users of AI to the ethical guidelines and moral values of how to build them and how to use them may be key to better awareness and ability to answer tough questions on obligations and responsibilities.

This study has recorded how from one generation to the next, the students’ perceptions of the safety of having a society immersed in artificially intelligent devices change over years. The change is significantly positive and pro technological advancement. Early in the study, students did not think that technological advancements in automation and artificial intelligence would be possible but thought it was worthwhile being careful, while in later years, students got excited by what they saw and heard about how far technological advancements had reached with

classic examples discussed in class such as Boston Dynamics robots jumping, dogs running, etc. They voiced not necessarily being worried about the ethics of such developments because they felt the responsibility comes with the development and so we “just” must be careful. Upon further discussion and debate using ethical frameworks such as Kantian’s goodwill and intention, or Utilitarian’s consequences, they accepted that the responsibility must be developed and understood and not taken for granted or implicit.

While this may be an expected outcome given how technologically immersed the new generations are, a significant finding of this study also shows that although students did seem more pro-artificial intelligence, more pro-autonomous devices, and pro-giving them more choices, in the journal-posts students ultimately voiced concerns as discussed in the class, showing a possible shift in their perceptions, questioning moral position of artificial intelligence. This highlights the importance of teaching ethics courses to students in engineering and IT. This also highlights a significant concern that students may not have fully grasped the implications of the advances being made or thought of in the future.

The findings of this study are significant to academics, researchers, program coordinators, tertiary education management, and policy makers because it is a step toward positing the urgency and need to take ethics in AI seriously primarily because the study shows how students possibly do not understand the full extent and potential of AI to do harm and the matter of bias in developing or using AI in manners that may be injurious to the economy, politics, or even society, thus proving the seriousness and urgency of having and running stand-alone ethics courses to computer and engineering students.

This study has tremendous future scope. The next stage of the research looks at the learning objectives, content, ethical frameworks, and assessments in order to propose a masterclass after developing an in-depth understanding of student perception of ethics of artificial intelligence using quantitative methods that can be used by academics.

References

1. C. Allen, W. Wallach, I. Smit, Why machine ethics? *IEEE Intell. Syst.* **21**(4), 12–17 (2006)
2. A. Arnold, Are millennials excited or threatened by AI? *Forbes* (2018) [Online] Available URL: <https://www.forbes.com/sites/andrewarnold/2018/03/25/are-millennials-excited-or-threatened-by-ai/#4d9456fc77e2> (last visited: 22/05/2020)
3. K.V. Bletzer, Visualizing the qualitative: making sense of written comments from an evaluative satisfaction survey. *J. Educ. Eval. Health Prof.* **12**(12) (2015). <https://doi.org/10.3352/jeehp.2015.12.12>. [Online] Available URL: <https://ieeexplore.ieee.org/document/7118241> Accessed 23 May 2020
4. N. Bostrom, E. Yudkowsky, The ethics of artificial intelligence. Ch 15, in *The Cambridge Handbook of Artificial Intelligence*, ed. by K. Frankish, M. Keynes, W. M. Ramsey, (Cambridge University Press, 2014), pp. 316–334
5. M.T. Chi, S.S. Lin, S.Y. Chen, C.H. Lin, T.Y. Lee, Morphable word clouds for time-varying text data visualization. *IEEE Trans. Vis. Comput. Graph.* **21**(12), 1415–1426 (2015). <https://doi.org/10.1109/TVCG.2015.2440241>

6. C. Clifford, Top AI experts warn of a 'Black Mirror' future swarms of micro-drones and autonomous weapons. CNBC (2018, 2020) [Online] Available URL: <https://www.cnn.com/2018/02/21/openai-oxford-and-cambridge-ai-experts-warn-of-autonomous-weapons.html>. Accessed 24 May 2020
7. C. Clifford, Bill Gates: AI is like nuclear energy – 'both promising and dangerous'. CNBC (2019, 2019) [Online] Available URL: <https://www.cnn.com/2019/03/26/bill-gates-artificial-intelligence-both-promising-and-dangerous.html>. Accessed 24 May 2020
8. A. Cramer, Artificial intelligence: The fourth industrial revolution. Information Age. Bonhill Group Plc (2018) [Online] Available URL : <https://www.information-age.com/artificial-intelligence-fourth-industrial-revolution-123475170/> Accessed 24 May 2020
9. V. Dignum, 2018. Ethics in Artificial Intelligence: Introduction to the Special Issue
10. M.M. Handelsman, A. Bashe, S.K. Anderson, Ethics of psychotherapy and counseling. In R. L. Miller, E. Balcells, S. R. Burns, D. B. Daniel, B. K. Saville, & W. D. Woody (Eds.), Promoting student engagement: Activities & demonstrations for psychology courses (Vol. 2, pp. 193–197) (2011). Syracuse, NY: Society for the Teaching of Psychology. Available from the STP website: <http://teachpsych.org/resources/e-books/pse2011/vol2/index.php>
11. A. Hern, Google whistleblower launches project to keep tech ethical. [online] the Guardian. (2019). Available at: <https://www.theguardian.com/world/2019/jul/13/google-whistleblower-launches-project-to-keep-tech-ethical>. Accessed 8 Oct 2019
12. Z.R. Khan, G. al Qaimari, S.D. Samuel, Information systems and technology education: Is education the bridge? in *Ch X in Information Systems and Technology Education: From the University to the Workplace*, ed. by G. R. Lowry, R. L. Turner, (IGI Global, 2007)
13. Z.R. Khan, S. Venugopal, 'E'ntries all the way using online reflective journal writing as innovative tool to enhance student understanding and performance in ethics courses for information age. *Int. J. Recent Technol. Eng.* **8**(2S11), 2829–2833 (2019)
14. R. Kruger, Discussing cyber ethics with students is critical. *Soc. Stud.* **94**(4), 188–189 (2003)
15. I.D. Maramba, A. Davey, M.N. Elliott, et al., Web-based textual analysis of free-text patient experience comments from a survey in primary care. *JMIR Med. Inform.* **3**(2), e20 (2015). <https://doi.org/10.2196/medinform.3783>. [Online] Available URL: <https://medinform.jmir.org/2015/2/e20/>. Accessed 23 May 2020
16. T. McConaghy, 3 ways we can maximize AI's impact on meeting the un sustainable development goals. Artificial intelligence. *Emerg. Trends. ICT4SDG. ITUNews.* [Online] (2019) Available URL: <https://news.itu.int/3-ways-we-can-maximize-ais-impact-on-meeting-the-un-sustainable-development-goals/>. Accessed 23 May 2020
17. M.S. McCormick, *Believing Against the Evidence: Agency and the Ethics of Belief* (2014). New York: Routledge
18. R. McMillan and D. Seetharaman, Facebook Finds Hack Was Done by Spammers, Not Foreign State. the Wall Street Journal. Published Oct 17 2018 (2018). [Online] Available URL: <https://www.wsj.com/articles/facebook-tentatively-concludes-recent-hack-was-perpetrated-by-spammers-1539821869>
19. K. Miller, T.P. Murphrey, Catching up with our students. Millennials and iGen: Is agriscience education ready? *Agric. Educ. Mag.* **83**(3), 20 (2010)
20. J.H. Moor, The nature, importance, and difficulty of machine ethics. *IEEE Intell. Syst.* **21**(4), 18–21 (2006)
21. J. Nnaji, Ethical issues in technology-mediated education. *Int. J. Cyber Ethics Educ. (IJCEE)* **2**(2), 44–51 (2012)
22. A.M. Ohme, H.K. Isaacs and D.W. Trusheim, 'Survey Participation: a study of student experiences and response tendencies' (2005), Office of Institutional Research and [Online] Available URL: www.udel.edu/IR/presentations/SurveyParticipation.ppt
23. R. Osmo and R. Landau, The Role of Ethical Theories in Decision Making by Social Workers, *Social Work Education*, 25:8, 863–876, (2006). <http://dx.doi.org/10.1080/02615470600915910>

24. D. Pegg, Facebook labelled 'digital gangsters' by report on fake news. *The Guardian*. Published 18 Feb 2019 (2019). [Online] Available URL: <https://www.theguardian.com/technology/2019/feb/18/facebook-fake-news-investigation-report-regulation-privacy-law-dcms>
25. J.C. Peña, L.A. García, The critical role of education in every cyber defense strategy. *N. Ky. L. Rev.* **41**, 459 (2014)
26. C.E. Rusbult, P.A. Van Lange, Why we need interdependence theory. *Soc. Personal. Psychol. Compass* **2**(5), 2049–2070 (2008)
27. S. Russell, S. Hauert, R. Altman, M. Veloso, Ethics of artificial intelligence. *Nature* **521**(7553), 415–416 (2015)
28. D.L. Schwartz and J.D. Bransford, A Time For Telling, *Cognition and Instruction*, 16:4, 475–5223, (1998). [10.1207/s1532690xci1604_4](https://doi.org/10.1207/s1532690xci1604_4)
29. E. Sobiesk, J. Blair, G. Conti, M. Lanham, H. Taylor, Cyber education: a multi-level, multi-discipline approach, in *Proceedings of the 16th Annual Conference on Information Technology Education*, (ACM, 2015), pp. 43–47
30. *The Economist*, (2018). Why is Mark Zuckerberg Testifying in Congress?. [online]. Available at: <https://www.economist.com/the-economist-explains/2018/04/09/why-is-mark-zuckerberg-testifying-in-congress>
31. Us.norton.com, (2019). 10 cyber security facts and statistics for 2018. [online] Available at: <https://us.norton.com/internetsecurity-emerging-threats-10-facts-about-todays-cybersecurity-landscape-that-you-should-know.html>. Accessed 8 Oct 2019
32. R.L. Verecio, Computer ethics awareness: Implication to responsible computing. *Int. J. Educ. Res.* **4**(3), 195–204 (2016)
33. R. Vinuesa, H. Azizpour, I. Leite, et al., The role of artificial intelligence in achieving the sustainable development goals. *Nat. Commun.* **11**, 233 (2020). <https://doi.org/10.1038/s41467-019-14108-y>
34. D. Whittier, Cyberethics in the Googling age. *J. Educ.* **187**(2), 1–86 (2007)
35. G. Yehiav, Paving the way for Gen Z in Tech. *Forbes Technology Council*. *Forbes* (2019). [Online] Available URL <https://www.forbes.com/sites/forbestechcouncil/2019/02/19/paving-the-way-for-gen-z-in-tech/#78e92f2165> (last visited 22/5/2020)

Predicting the Academic Performance of Undergraduate Computer Science Students Using Data Mining



Faiza Khan, Gary M. Weiss, and Daniel D. Leeds

1 Introduction

The field of data mining is concerned with finding relevant and meaningful patterns within a dataset. A dataset contains instances of data from various attributes, or factors. An instance contains the values of the attributes for one student, therefore, there were 82 instances in this dataset as 82 computer science students at Fordham University submitted the anonymous survey. The survey contained 23 questions related to the student's demographics, lifestyle, etc., and also asked for the student's GPA. Predictive models were built using a variety of data mining methods to identify the key features that impact student performance. By observing meaningful patterns within this student GPA dataset, we can determine which of the 23 factors are most useful in predicting student GPA and differentiating between below average, average, good, and great students. The design of the survey was influenced by the related work in this field, which suggested some factors that can impact student performance.

Sleep is one of the most influential factors of student GPA. Suffering from sleep deprivation and poor sleep quality negative impacts the academic performance of students [1–5]. A study shows that school-age students who were not sleep deprived were healthier and had higher IQ and perceptual reasoning overall as measured by the WISC-IV, a test to measure intellectual ability of children [2]. In another study, among 144 medical students, sleep quality of the students prior to the pre-clinical examination affected their result and final GPA [3]. Another study has results which indicates that students with the highest GPA had earlier bedtimes and earlier wake times [4]. Thus, the number of hours of sleep a student receives per night as well as

F. Khan (✉) · G. M. Weiss · D. D. Leeds

Dept. of Computer and Information Sciences, Fordham University, Bronx, NY, USA

e-mail: fkhan32@fordham.edu; gaweiss@fordham.edu; dleeds@fordham.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_21

303

the time that they typically go to sleep can be good indicators in predicting student GPA.

Moreover, sleep deprivation most prominently affects functional connectivity involving prefrontal areas [5]. Because the prefrontal cortex is responsible for decision making, students who are sleep deprived have less activity in the prefrontal cortex, which can make it harder for them to make decisions during tests and obtain a decent GPA. Furthermore, other factors may also affect one's sleep quality, such as their outlook on life [6]. People who are optimistic may fall asleep faster than those who are pessimistic. Because the studies that were previously mentioned showed that good sleep quality resulted in higher GPA and people who are optimistic tend to have better sleep quality, it can be concluded that students who are optimistic generally have higher GPA than students who are pessimistic.

Drug use can also be detrimental to academic performance. Research shows that alcohol and drug use among college students resulted in lower GPA's than those who used such drugs minimally or refrained from drug use [7]. The prefrontal cortex is also affected by drug use, especially among teens and young adults since this region of the brain develops until the mid 20s. Damage to the prefrontal cortex from substance abuse, therefore, can result in poor performance.

The number of hours spent studying, social media use, and the student's personality may also influence their GPA. Students who study for longer periods of time are more likely to be prepared for exams than students who do not dedicate as much time to their studies. Because the overall grades for undergraduate computer science courses are primarily determined by midterm and final examination scores, performing well on exams results in a higher GPA among undergraduate students. Moreover, increased use of social media may result in less time dedicated towards studying. In turn, this can lead to a lower GPA. Personality is another factor which can affect GPA since studies show that introverts tend to be better listeners than extroverts, [8] which signifies why this feature was another good indicator determining student GPA on some classifiers. Undergraduate courses are typically lecture-based, thus listening attentively and taking good notes can help students better prepare for exams, and in turn, earn a higher GPA.

The next section, Sect. 2, covers experimental methodology which explains the approach to this classification task and how the performance of each of the classifiers were evaluated. The results and analysis the performance of the classifiers on this student GPA dataset are covered in Sect. 3. Section 4 mentions related work in this field and the concepts that future studies can focus more on are discussed in Sect. 5.

2 Experiment Methodology

2.1 Attributes Used in the Survey

82 undergraduate computer science students at Fordham University were given a survey which they filled out anonymously. The survey contains 23 questions, and the responses were used to predict GPA (Table 1).

Table 1 Name and description of attributes

	Feature name	Description
1	Gender	Male, female, or other
2	Age	Student’s age
3	Year	Year in school
4	Major	All students were Computer Science majors
5	Race	Student’s race
6	Ethnicity	Hispanic or Not Hispanic
7	Sleep	Hours of sleep per night
8	socialMedia	Hours of social media use per week
9	Studying	Hours of studying per week
10	ResComm	Campus resident or commuter
11	studIncome	Student’s annual income
12	parIncome	Total household annual income
13	Job	Whether or not the student currently has a job
14	Drugs	Illicit drug use on a scale of 1–5
15	Alcohol	Alcohol use on a scale of 1–5
16	Physical	Hours of physical activity per day
17	Notes	Prefer taking notes with notebook or laptop
18	OlderSiblings	Number of older siblings
19	YoungerSiblings	Number of younger siblings
20	Outlook	Pessimistic, optimistic, or neutral outlook on life
21	Personality	Introverted or extroverted
22	Satisfied	Satisfied with academic grades
23	classGrades	GPA (below average, average, good or great)

The number of hours spent studying per week, attribute 9, was a multiple choice question for students in which 1 represents studying for 0–5 hours per week, 2 represents studying for 6–10 hours per week, 3 represents studying for 11–15 hours per week, 4 represents studying for 16–20 hours per week, and 5 represents studying for more than 20 hours per week.

Illicit drug use and alcohol use, attribute 14 and 15 respectively, were based on a scale of 1–5 in which 1 signifies never used, 2 signifies rarely used, 3 signifies used every other week, 4 signifies used every week, and 5 signifies everyday use.

Total annual household income, attribute 12, was a multiple choice attribute for students in which 1 represents \leq \$30k, 2 represents between \$30k and \$50k, 3 represents between \$50k and \$70k, 4 represents between \$70k and \$100k, 5 represents between \$100k and \$350k, and 6 represents \geq \$350k.

Below are attribute distributions of some of the factors which will be analyzed in Sect. 3. Table 2 shows the attribute distribution of illicit drug use and Table 3 shows the attribute distribution of the number of hours spent on social media per week. Table 2 indicates that almost two-thirds of the students in this dataset reported never using illicit drugs. Table 3 shows that more than one-third of the students in the

Table 2 Illicit drug use attribute value distribution

Illicit drug use frequency	Number of students
1 (Never used)	51
2 (Once a month/rarely)	16
3 (Sometimes)	5
4 (Every week)	8
5 (Almost everyday)	2

Table 3 Hours spent on social media per week

Social media use	Number of students
0–5 hours per week	32
5–10 hours per week	20
10–15 hours per week	13
15–20 hours per week	7
20+ hours per week	10

Table 4 GPA class distribution

Academic performance category	Number of students
Below average	5
Average	14
Good	26
Great	35

dataset use social media for less than five hours per week, while the rest of the students use social media for longer periods of time.

Table 4 shows the GPA class distribution. Students at Fordham University have a GPA between 0.0 and 4.0. In order to convert this regression task into a classification task, the corresponding class values were assigned to the following GPA scores: poor (0.0–2.49), below average (2.50–2.99), average (3.0–3.32), good (3.33–3.66) and great (3.67–4.0). A 4.0 is the highest possible GPA a student can obtain at Fordham University. No student reported a GPA under 2.50, hence the “poor” category does not appear in this dataset. There were 82 total instances in the dataset, however, 2 of those instances do not contain the class value. Hence, the performance of the algorithms was determined by the 80 instances which have the class value (student GPA).

2.2 Data Mining Algorithms Used

The classifiers used on this student GPA dataset include Nearest Neighbor, Decision Trees, Random Forest, Random Trees, and Multilayer Perceptron. The performance of these data mining algorithms were compared against the performance of ZeroR, the baseline classifier which always predicts the majority class value of a dataset. For all algorithms, the default parameters on the WEKA data mining application are used to ensure fairness in performance on this dataset.

IBk Nearest Neighbor

Nearest Neighbor is an instance-based classifier which classifies an unseen instance based on its distance to other training records. The distance to the training record determines the level of similarity, thus an unseen instance is classified into the same category as its closest trained instance due to their similar attribute values. The default parameter uses one training record as a Nearest Neighbor (also known as 1-Nearest Neighbor), hence the class label of the closest training instance determines the class value of that unseen instance.

J48 Decision Tree

A Decision Tree classifier splits the data based on the attributes that result in the lowest entropy in an effort to maintain homogeneity. Each leaf node in the Decision Tree corresponds to a combination of rules, or factors, which resulted in that classification.

Random Forest

Random Forest is a computationally fast algorithm which creates an ensemble of Decision Trees. Ensembles have more expressive power and can assist with bias and variance by averaging over multiple runs. The final decision is made via majority voting of the trees in the forest in the Random Forest method.

Random Tree

Similar to the Decision Tree algorithm, the Random Tree algorithm generates an output in the structure of a tree that is easy to understand and justify. It also employs the method of bagging to produce a random set of data for constructing a Decision Tree and generates many individual learners.

Multilayer Perceptron

Multilayer Perceptron is a class of artificial neural networks. This algorithm leverages the use of hidden layers for inputs, and it assigns weights to the attributes based on usefulness in classification of instances. These weights are typically not easy to justify or explain. The weights of the same attribute can vary among the different sigmoid nodes in this method. Sigmoid nodes are used in backpropagation with the associated data in the Multilayer Perceptron algorithm. Each sigmoid node contains the attributes and their associated weight.

2.3 Evaluation Metrics

A training set contains instances which serves as input to the data mining algorithm while the test set contains instances which must be classified by the algorithm. Classifiers build a model on the training set and they are evaluated based on their performance on the test set. 10-fold cross-validation, the partitioning method used in this experiment, is a type of cross-validation which entails partitioning a dataset into 10 partitions, training on 9 partitions and testing on the remaining section, iterating for 10 times. To generate a test set, the 10-fold cross-validation method is used so that every instance in the training set can be a part of the test set. The results of the 10-fold cross-validation on the different algorithms will be analyzed in Sect. 3.

There is also minimal preprocessing involved for this experiment. The preprocessing only consists of the removal of ID (an attribute which was assigned to each of the surveys to identify the instances). Because many of the attributes in combination with one another proved to be useful in predicting student GPA, no additional attributes were removed in the preprocessing stage.

To emphasize the importance of a large dataset and to analyze how the predictive accuracy changes based on a smaller sample size, the algorithms are also tested using 10-fold cross-validation on 25%, 50%, and 75% of the data. As the size of the dataset decreased, the algorithms generally performed worse.

3 Results

In this section, the performance of the following algorithms using 10-fold cross-validation are discussed. Random Forest had the highest predictive accuracy on this dataset. The ZeroR (baseline) classifier performs the worst, as depicted in Table 5.

As mentioned previously, this classifier predicts the majority class value. Because the majority of the students in the dataset reported having a great GPA (35 out of 80 students), the baseline always predicts that the students have a great GPA. The table indicates that Nearest Neighbor, Decision Trees, Random Forest, Random Tree, and Multilayer Perceptron were able to learn the dataset well since their predictive accuracy was significantly higher than the performance of the baseline classifier. The performance of those algorithms yields useful information regarding the GPA classification of undergraduate computer science students.

Table 5 Accuracy among different classifiers

Classifier	Nearest neighbor	Decision tree	Random forest	Random tree	Multilayer perceptron	ZeroR (baseline)
Accuracy	91.25%	75.0%	95.0%	82.5%	90.0%	43.75%

3.1 Analysis of the Performance of IBk Nearest Neighbor

The accuracy rate for Nearest Neighbor algorithm on this dataset is 91.25%, which shows that there are similar characteristics among students with the same GPA classification. Instances that are a part of the training set may have similar attribute values to unseen instances in the test set. The high accuracy rate indicates that the Nearest Neighbor algorithm learned this dataset well. Using 1-Nearest Neighbor (the default parameter of Nearest Neighbor) shows that students within the same GPA classification typically shared similar attribute values.

3.2 Analysis of the Performance of J48 Decision Tree

The accuracy rate for Decision Tree on this dataset is 75.0% and it was the poorest performing algorithm among the 5 classifiers discussed in this section. Decision Trees typically handle irrelevant features well, but perhaps the attribute values used to split the data did not yield the lowest entropy. Complex rules cannot be expressed well with Decision Trees, and because there are several factors which can be used to predict student GPA in combination with one another, the Decision Tree was not able to perform very well on this dataset (Fig. 1).

The output of the Decision Tree suggests that the number of hours of sleep and illicit drug use are the two most important factors for classification of student GPA in this dataset since these are the first two splits in the tree. Sleep is the first split, and students who reported sleeping less than 4 hours per night were automatically categorized as having a “below average” GPA. This signifies that sleeping more hours per night may increase the likelihood of a student performing well academically. Among students who sleep more than 4 hours per night, those who use illicit drugs generally have a low GPA classification whereas students who reported never using drugs are classified as having a good or great GPA. Students who use drugs are not very likely to have a great GPA in this dataset since only one leaf node in the drugs greater than 1 subtree contains instances classified as “great.”

The third split in the Decision Tree is either the number of hours spent studying or student age, depending on which subtree of the drugs feature the student belongs to. For students who never used illicit drugs, the next feature that the Decision Tree splits on is the number of hours spent studying. As shown in the Decision Tree Output, students who sleep more than 4 hours per night, have never used drugs, and study more than 15 hours per week were all classified as having a great GPA. This particular rule applied to 13 instances, signifying that more than one-third of the students who had a great GPA were classified as “great” based on the rule which combines these three factors. For students who have used illicit drugs, the Decision Tree splits on age. All students who sleep more than 4 hours per night, use drugs, and are 20 years old or younger are classified as having an average GPA, which is the second lowest classification in this dataset (Fig. 2).

```

sleep <= 4: belowAverage (5.0/1.0)
sleep > 4
|
|   drugs <= 1
|   |
|   |   studying <= 3
|   |   |
|   |   |   personality = introverted
|   |   |   |
|   |   |   |   race = AmericanIndian: great (0.0)
|   |   |   |   race = Asian: great (6.0)
|   |   |   |   race = AfricanAmerican: good (2.0)
|   |   |   |   race = NativeHawaiian: great (0.0)
|   |   |   |   race = White
|   |   |   |   |
|   |   |   |   |   gender = M
|   |   |   |   |   |
|   |   |   |   |   |   studIncome <= 7000: good (5.0)
|   |   |   |   |   |   studIncome > 7000: great (2.0)
|   |   |   |   |   |
|   |   |   |   |   |   gender = F: great (7.0)
|   |   |   |   |   |   gender = 0: great (0.0)
|   |   |   |   |   |
|   |   |   |   |   |   race = Other: good (1.0)
|   |   |   |   |
|   |   |   |   |   personality = extroverted
|   |   |   |   |   ResComm = resident
|   |   |   |   |   |
|   |   |   |   |   |   alcohol <= 2: great (3.0)
|   |   |   |   |   |   alcohol > 2: good (2.0)
|   |   |   |   |   |
|   |   |   |   |   |   ResComm = commuter: good (8.0)
|   |   |   |   |
|   |   |   |   |   studying > 3: great (13.0)
|   |
|   |   drugs > 1
|   |   |
|   |   |   age <= 20: average (12.0/1.0)
|   |   |   age > 20
|   |   |   |
|   |   |   |   alcohol <= 3: average (2.0)
|   |   |   |   alcohol > 3
|   |   |   |   |
|   |   |   |   |   ResComm = resident
|   |   |   |   |   |
|   |   |   |   |   |   year = freshman: great (0.0)
|   |   |   |   |   |   year = sophomore: great (0.0)
|   |   |   |   |   |   year = junior: good (2.0)
|   |   |   |   |   |   year = senior: great (4.0)
|   |   |   |   |   |
|   |   |   |   |   |   ResComm = commuter: good (6.0)

```

Fig. 1 Decision tree output

Fig. 2 Confusion matrix for decision tree

===Confusion Matrix===				
a	b	c	d	← classified as
4	1	0	0	a = belowAverage
1	11	2	0	b = average
0	0	16	10	c = good
1	0	5	29	d = great

One student who had great GPA was classified as having a below average GPA based on the Decision Tree confusion matrix, which demonstrates that this algorithm was off by three classes for only one particular instance. For all other instances, the Decision Tree either classified the GPA correctly, or misclassified by only one class value. Additionally, some students with good GPA were classified as “great” and some students with great GPA were classified as “good.” This means that the attribute values for students with a GPA of 3.33 or above were more similar

than the attribute values for students with significantly lower GPA. Because some of the responses for these two groups of students were similar, some attributes of the Decision Tree split did not provide low entropy for the classification of these student GPA's.

3.3 Analysis of the Performance of Random Forest

The Random Forest classifier used an ensemble of Decision Trees in order to classify student GPA, and it resulted in the highest accuracy rate among all of the algorithms discussed in this paper. The accuracy rate for Random Forest is 95.0%, which is significantly higher than the accuracy rate for Decision Tree. This suggests that the Random Forest algorithm was able to learn the dataset extremely well (Fig. 3).

While this algorithm performed the best among all classifiers, it categorized one student with a below average GPA as "great". These two classes are on opposite ends of the spectrum as the lowest GPA classification is below average (2.50–2.99) and the highest GPA classification is great (3.67–4.0) in this dataset. However, the classification accuracy is significantly higher for this classifier since only 4 instances were categorized inaccurately.

3.4 Analysis of the Performance of Random Tree

The accuracy rate for Random Tree on this dataset is 82.5%. Because Random Tree performed better than Decision Tree for this dataset, the attribute values which were used to split the data in Random Tree most likely resulted in lower entropy for more accurate classification.

The output of the Random Tree classifier suggests that the number of hours spent studying per week, total annual household income, and the number of hours of physical activity per day are the three most important factors in the classification of student GPA since these are the first three splits in the tree. The Random Tree first splits on the number of hours spent studying per week. Students who studied less than 15 hours a week typically had lower GPA classifications, depending on other factors. Next, the Random Tree splits on either total household income or

Fig. 3 Confusion matrix for random forest

===Confusion Matrix===				
a	b	c	d	← classified as
4	0	0	1	a = belowAverage
0	14	0	0	b = average
0	0	24	2	c = good
0	0	1	34	d = great

the number of hours of physical activity per day, depending on which subtree the student belongs to. Students who studied more than 15 hours per week but exercised for less than three-quarters of an hour (45 minutes) per day typically had lower GPA classifications than students who studied more than 15 hours per week and exercised at least 45 minutes per day. Thus, exercising regularly can improve the academic performance of students. More than one-third of the students who were classified as having a great GPA reported studying more than 15 hours per week and exercising at least 45 minutes per day.

Additionally, for students who study less than 15 hours per week, the next attribute that the Decision Tree splits on is the income attribute. If the total household annual income exceeds \$100k (value of 4 or greater on the scale of 1–6), then the student is more likely to achieve either a good or great GPA if they use social media for less than 17.5 hours per week and use illicit drugs either rarely or never. In contrast, students who reported having a household income greater than \$100k but used social media for more than 17.5 hours per week generally had lower GPA classifications. In fact, while many of the leaves in the income greater than \$100k subtree correspond to having a great or good GPA, students who spent more than 17.5 hours on social media per week and slept for less than 5.5 hours per night were all classified as having below average GPA, which is the lowest classification. Therefore, an increased number of hours spent on social media, combined with other factors, can negatively impact GPA for undergraduate computer science students. Furthermore, students can still be classified as having a great GPA with a household income lower than \$100k if they are optimistic and use social media platforms for less than 4.5 hours per week. This suggests that while household income can be an important factor for predicting academic performance, other factors such as social media use and drug use can determine the final classification of student GPA.

The split on the outlook attribute is interesting because some leaf nodes corresponded to a great GPA for optimistic and neutral outlooks, but no leaf nodes corresponded to a great GPA for a pessimistic outlook. This shows that a positive or neutral outlook on life can help undergraduate computer science students obtain a higher GPA.

Additionally, the frequency of drug use also impacted GPA. All 5 students who were in the lowest GPA classification (below average) reported moderate to high frequency of illicit drug use (attribute value was 3 or higher on the scale of 1–5). Therefore, drug use negatively impacts student GPA.

It is important to note that the Random Tree and Decision Tree output generates some rules which may not be outputs of a larger student GPA dataset since some factors may not be causally linked and may simply be co-occurrences of one another. Additionally, the Random Tree also splits on the same attribute within its subtrees sometimes, unlike the Decision Tree output which splits on each attribute only once in each subtree. For instance, there were several splits in the Random Tree on attributes such as the number of hours spent studying per week and number of hours spent on social media platforms. This suggests that even minor increases in the number of hours spent on social media can negatively impact undergraduate student GPA.

3.5 Analysis of the Performance of Multilayer Perceptron

The Multilayer Perceptron algorithm had an accuracy rate of 90.0% on the classification of student GPA for this dataset. It may be difficult to justify the weights of the Multilayer Perceptron algorithm, however, there were 25 sigmoid nodes (which are nodes with different weights on features) in which the highest weights were usually assigned to the following attributes: studying, drugs and parental income. This may signify that these values were the most useful in predicting student GPA for this classifier.

3.6 General Analysis

Below are some more general observations regarding the output of these algorithms:

- Students who slept less than 4 hours per night have lower GPA as suggested by the J48 Decision Tree Algorithm. Additionally, students who slept more than 8 hours (in combination with other factors) were classified as having great GPA by the Random Tree output.
- Students who studied more than 15 hours per week, have never used illicit drugs, and typically slept for more than 4 hours per night were classified as having a great GPA.
- Undergraduate seniors can be classified as having a great GPA in the Decision Tree even if they drink alcohol at least once a week and use illicit drugs. This suggests that those who have reached the legal age of drinking might be impacted less severely than those who are younger.
- Students who are older tended to perform better academically than younger students. This may occur because computer science students can enhance their knowledge as they gain more experience in their field of study. As a result, upperclassmen may perform better in their courses and obtain a higher GPA than freshmen and sophomores.
- Students who spent 17.5 or more hours per week on social media tended to study less than 15 hours per week, and as a result, had lower GPA's than those who dedicated more time to their studies.

3.7 Relationship Between Features and the GPA Class Value

Below are the graphs which depict the relationship between the frequency of illicit drug use and GPA (Fig. 4a), the number of hours spent on social media platforms per week and GPA (Fig. 4b), and the number of hours of studying per week and

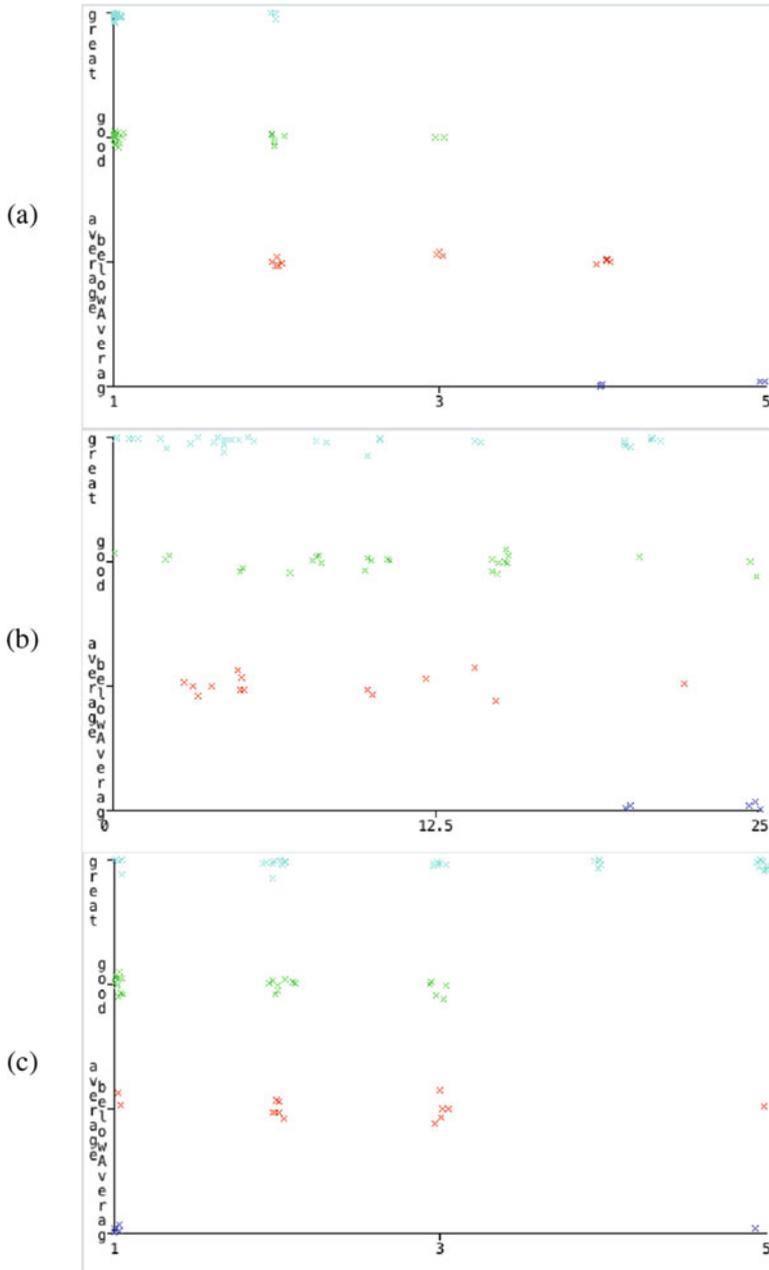


Fig. 4 Relationship between feature and GPA. (a) Illicit drug use frequency. (b) Hours of social media use per week. (c) Hours spent studying per week

GPA (Fig. 4c). The purple dots represent students belonging to the “below average” GPA class, the red dots represent students belonging to the “average” GPA class, the green dots represent students belonging to the “good” GPA class, and the blue dots represent students belonging to the “great” GPA class.

Figure 4a shows that as the frequency of drug use increases, students typically have lower GPA’s. Most of the students with good or great GPA reported never using illicit drugs. Figure 4b shows that while it is possible to obtain a great GPA when using social media for more than 12.5 hours per week, most students with great GPA use social media platforms for fewer hours. Figure 4c shows that as the number of hours spent towards studying per week increases, students are generally able to achieve a higher GPA classification. Most students with a great GPA studied for more than 15 hours per week (indicated by 3 or higher as shown in the graph).

3.8 Results Obtained by Reducing the Size of the Dataset

The size of the dataset was reduced to observe how the algorithms perform using 75% of the dataset, 50% of the dataset, and 25% of the dataset. As the number of instances decrease, the algorithms generally perform worse (Table 6). The original, full dataset has the highest accuracy rates since the algorithms had enough instances to learn the data and provide useful information regarding classification of student GPA.

The algorithms were not able to learn the data well when the number of instances were reduced. More training data generally improves classifier performance. Perhaps if the original, full dataset contained more instances, the accuracy rates would have been much higher since the algorithms would have more learning data. Additionally, even if the accuracy rates were higher after reducing the size of the dataset, we cannot generalize such information as it may not apply to a larger population of undergraduate students. The more instances in the dataset, the less generalization will occur.

Table 6 Accuracy of reduced size datasets on 10-fold cross-validation

Training size	Nearest neighbor	J48 decision tree	Random forest	Random tree	Multilayer perceptron
75% of dataset	78.3%	63.3%	83.3%	73.3%	80.0%
50% of dataset	52.5%	45.0%	57.5%	40.0%	62.5%
25% of dataset	40.0%	45.0%	50.0%	45.0%	55.0%

4 Related Work

This section contains discussion regarding some related works of researchers who studied the classification of student GPA using other datasets.

A study aimed to predict the students' final GPA based on the dataset performance on Decision Trees [9]. Based on student grades on previous courses, such as computer architecture, computer ethics and software engineering, the researchers predicted the student final GPA as average, good, very good, and excellent. The predictive accuracy of the models were not presented in this paper, but the researchers demonstrate how GPA results from previous courses can be used to predict student GPA for future semesters. The attribute values used for this classification task include student grades from introductory courses such as software engineering, computer architecture, Java1, etc.

Another study uses feature extraction to classify students based on their academic performance [10]. The researchers primarily focused on the classification of students who have poor academic performance. They extracted features from historical grading data in order to test different simple and sophisticated classification methods based on big data approaches. Gradient Boosting and Random Forest performed the best for this experiment. To classify level A students, the highest accuracy was obtained when using course background and prerequisite attributes. To classify students who have failing GPA, the researchers examined specific reasons which are related to the individuals themselves, not the class background, prerequisite, or similar courses. Area under the receiver operating characteristic (ROC) curve for Gradient Boosting was the highest with a value of 0.877.

Alcohol and drug use can also be significant factors in predicting student GPA as shown in various studies. A study shows that high drug use leads to more absence from school, which affects students' overall academic performance [11]. The results from the dataset demonstrate that girls were more likely than boys to try alcohol and boys were more likely to try illicit drugs. Both illicit drugs and alcohol affect student GPA according to the researchers as increasing levels of drug and alcohol consumption were associated with lower GPA and a higher number of days and hours missed from school [11]. Logistic Regression was the data mining algorithm that was used by these researchers, but the accuracy results of this method are not stated. According to the researchers, some attributes were not helpful in predicting student GPA since girls typically had higher GPA than boys, but also had more missed days from school on average. Thus, days missed from school was not as important as the student's alcohol and drug use for this dataset.

5 Conclusion

In this paper, the features which were most important in predicting the GPA of undergraduate computer science students in this dataset were analyzed. The output of the data mining algorithms suggest that a combination of factors such as the

number of hours of sleep per night, the frequency of illicit drug use, the number of hours spent studying per week, and the number of hours of social media use per week provide useful information that can be used to successfully classify the GPA of computer science majors. Students who slept for more hours per night, studied for more than 15 hours per week, spent less time using social media per week, and never used illicit drugs tended to have the highest classification of GPA. The Random Forest algorithm performed the best among all classifiers with a 95.0% accuracy rate, but the other algorithms still performed much better than the baseline classifier. Therefore, students with the same classification of GPA tend to have similar characteristics.

Future studies can integrate some concepts from the attributes that were used in this dataset and also focus on how specific kinds of drugs affect GPA since some drugs might be more detrimental to academic performance than others. Researchers can investigate how other factors impact GPA, such as the number of hours per week spent towards self-care or meditation. Future studies can also increase the sample size as doing so can result in higher accuracy rate and yield useful information pertaining to student performance. One limitation of this dataset is that there might not be enough instances to generalize the findings to a larger population of undergraduate computer science students.

References

1. G. Curcio, M. Ferrara, L. De Gennaro, Sleep loss, learning capacity and academic performance. *Sleep Med Rev.* **10**(5), 323–337 (2006)
2. R. Gruber, R. Laviolette, P. Deluca, E. Monson, K. Cornish, J. Carrier, Short sleep duration is associated with poor performance on IQ measures in healthy school-age children. *Sleep Med.* **11**(3), 289–294 (2010)
3. K. Ahrberg, M. Dresler, S. Niedermaier, A. Steiger, L. Genzel, The interaction between sleep quality and academic performance. *J Psychiatr Res.* **46**(12), 1618–1622 (2012)
4. A.H. Eliasson, C.J. Lettieri, A.H. Eliasson, Early to bed, early to rise! Sleep habits and academic performance in college students. *Sleep Breath.* **14**, 71–75 (2010)
5. I.M. Verweij, N. Romeijn, D.J. Smit, et al., Sleep deprivation leads to a loss of functional connectivity in frontal brain regions. *BMC Neurosci.* **15**, 88 (2014)
6. Harvard Health Publishing, Harvard Medical School. Positive outlook may mean better sleep <https://www.health.harvard.edu/staying-healthy/positive-outlook-may-mean-better-sleep>
7. PLOS Research News, Alcohol and marijuana use associated with lower GPA in college. <https://researchnews.plos.org/2017/03/08/alcohol-and-marijuana-use-associated-with-lower-gpa-in-college/>
8. Science Daily, Who learns foreign language better, introverts or extroverts? <https://www.sciencedaily.com/releases/2017/07/170721104246.htm>
9. M.A. Al-Barrak, M. Al-Razgan, Predicting students final GPA using decision trees: A case study. *Int. J. Inf. Educat. Technol.* **6**(7), 528–533 (2016)
10. A. Polyzou, G. Karypis, Feature extraction for classifying students based on their academic performance, in *Proceedings of the 11th EDM Conference* (2018)
11. O. Heradstveit, J.C. Skogen, J. Hetland, M. Hysing, Alcohol and illicit drug use are important factors for school-related problems among adolescents. *Front. Psychol.* **8**, 1023 (2017). <https://doi.org/10.3389/fpsyg.2017.01023>. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5476929/>

An Algorithm for Determining if a BST Node's Value Can Be Changed in Place



Daniel S. Spiegel

1 Introduction

Part of graph theory, a *tree* is a connected graph that contains no cycles [7]. A connected graph is one where every node is coincident with at least one edge. A tree consists of a set of nodes and a set of directed edges, where each edge connects a pair of nodes.

The trees referred to in this work are *rooted*, meaning that:

- One node r is designated as the *root*
- Every other node c in the tree is connected by a single edge from one other node p
 - p is denoted as the parent of c , and c is denoted as a *child* of p .

The depiction of a general tree is ordinarily vertically oriented with the root node on top. All nodes are connected to the root by one or more edges and are depicted vertically in levels below the root, where level l consists of all nodes that are reachable by traversing l edges from the root.

A *binary tree* is a tree where a node can have no more than two children, i.e., any node can have zero, one, or two children. Each node has a left and a right *subtree*, although one or both can be empty. The specific binary tree of interest for this work is a *binary search tree* (BST), which is ordered according to its data. The rules for a BST apply to every node n in a BST:

- The data in every node in the left subtree is less than (based on the interpretation of “less than” for the data type) the data in n .

D. S. Spiegel (✉)

Kutztown University of PA, Kutztown, PA, USA

e-mail: spiegel@kutztown.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_22

319

- The data in every node in the right subtree is greater (based on the interpretation of “greater than” for the data type) than the data in n .

This invariant strictly orders the data and results in a BST being an extremely efficient container for storing data ordered alphanumerically, according to the data, with $O(\log_2 n)$ insertion and search algorithms. Insertion of new data always occurs with the placement of a new *leaf node* in the tree. Leaves are found at the lowest levels of a tree; they have no children.

Insertion occurs following a search, where the tree is negotiated according to the value of the new data to insert. Starting from the root, the new data is compared with the data in the current node. If the new data is less than the current node’s data, the search goes to the left child of the current node. Otherwise it goes to the current node’s right child. When a null pointer is encountered, the new data is placed in a node that is a child of the previous node, in the direction followed where the null pointer was encountered.

This methodology is also followed for a search for data. The difference is that a successful search terminates before encountering a null pointer, if the data is matched. The manner of providing information that the search was successful (or not) is outside the scope of this work.

Printing traversals, as for any container, are $O(n)$. Deletion from a BST has several cases based upon the position of data within the tree, i.e., whether the node with the data that is to be deleted has zero, one, or two children. Any data structures textbook, such as Weiss [7], has a detailed description of the deletion process, which is omitted here, as it is again outside the scope of this work.

A BST algorithm that doesn’t commonly appear in literature is editing (or changing) a value already stored in a BST. A naïve solution [8] is to delete the node and reinsert the updated data, which is placed in a new node. But there is no consideration of whether the updating of the data can be done in place, i.e., whether updating the data can be accomplished in place without invalidating the tree invariant. Deleting and reinserting data is inefficient if updating the data can occur in place while maintaining the tree invariant.

In this work, several tree applications will be briefly discussed, followed by a description of an algorithm that can be employed to determine whether a value in a BST can be changed/edited in place while maintaining the tree invariant, or property, that all elements in a node’s left subtree are lesser and all elements in a node’s right subtree are greater. The efficiency of the algorithms will be compared to the naïve deleted and reinsert method, and finally, conclusions will be drawn.

2 Tree Applications

Tree structures are used in many areas to implement real-world applications and this goes back decades. For example, in Casey [1] a method for using a tree search to implement queries in a data collection is described. Binary Indexed Trees,

introduced in Fenwick [2], maintains numerical sequences on which operations can be carried out using tree-style operations, of course with time complexity of $O(\log_2 n)$.

Also, of historical significance are tree representations of expressions. Redziejowski [5] describes the creation of trees to hold arithmetic expressions in such a manner as to minimize the number of required accumulators. Later, Genetic programming was an evolutionary technique that employed tree structures, as described in O'Reilly [4].

One of the most prominent tree applications, while not specifically using binary trees, is found in genealogy. Shockley [6] described manipulating genealogical data for optimal storage via tree structures. The use of tree structures to optimize memory use was mandatory at a time when the cost of memory was astronomical and its availability in any digital computer quite limited, particularly relative to current times. Kluge [3] discussed traversal of tree structures in memory using shift registers. This was necessary to maintain computing time within reasonable bounds.

3 Algorithm to Determine Whether a Change Can Occur in Place

In order to determine whether a change/edit can occur in place in a BST, the closest values, greater and less than, to the value to be affected must be determined. Once that has occurred, if the new value to be placed in the node is between the values closest to the current value, the change/edit can occur in place.

Thus, the algorithm must provide a method for determining the closest values. The algorithm makes the determination of neighboring values based on a node's relative location.

3.1 Relative Location Algorithm

This algorithm determines closest values by considering a node's children and location. While the deletion method has its greatest complexity when a node has two children, this algorithm is most direct in the same situation.

Node to Be Changed Has Two Children

Consider the BST in Fig. 1. In this situation, if b is the value to be changed, the two values closest to b must be d and e . This is because by the BST invariant, all nodes in a 's left subtree must be less than a . It follows that e is bigger than b and less than a . We can therefore conclude that if the value to which b will be updated is between

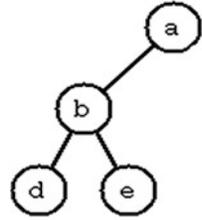
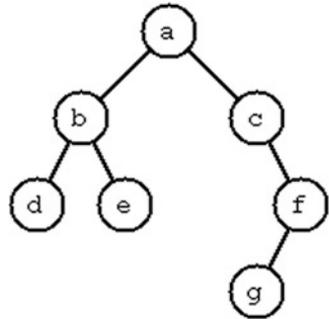
Fig. 1 Change b 

Fig. 2 Small BST



d and e , then the change can be in place, with no need to delete and reinsert any nodes.

The general case of whether a node with two children can be changed in place is a bit more involved, and it harkens back to deleting a node with two children. First, it must be noted that the value held in a node with two children does not always have as its closest values those held in those children, respectively.

Next, in the node deletion algorithm, deletion of a node with two children is accomplished by copying one of the closest values, which are found as either the rightmost value in the left subtree or the leftmost value in the right subtree, and then deleting the node whose value was copied, as it can't possibly have two children.

In Fig. 2, if a is to be deleted, it would be replaced with the data in either e or c , and that node would be deleted.

Note that in the left subtree, b and d are less than e , and in the right subtree c is less than f and g . The extreme in a subtree is found by going in that direction for only one step and then the opposite way as far as possible. In the right subtree of a in Fig. 2, the move to the right gets to c and no move the opposite direction is possible. That leaves c as the smallest value in that subtree. Similarly, e is the largest value less than a , the farthest right value encountered after taking one step left from a .

Thus, a can be changed in place if it falls between e and c .

Node to Be Changed Has No Children

The BST in Fig. 2 has three leaf nodes, i.e., nodes with no children, d , e , and g . To determine whether each can be changed in place requires a unique process. This is because of their location in the tree. Each will be described:

- d is an extreme value, i.e., it is the smallest value in the tree. It can be changed to any value less than b .
- e 's value falls between b and a . Any edit may not change it to a value outside range $[b,a]$ if it is to be changed in place.
- g 's value falls between c and f . Any edit may not change it to a value outside range $[c,f]$ if it is to be changed in place.

Visual identification of the situations is straightforward, but an algorithmic method of identifying the neighbors closest in value must be designed. For the three values noted, it can be observed that their closest value can be discerned according to the direction(s) traveled in the depth-first search required to arrive at their respective nodes.

What is notable about the path from the root to d is that every move between levels was in one direction. Such a path to a leaf can only lead to an extreme value in the tree. Note also the path from the root to f , which must be the largest value in the tree. f 's node has a left child but is still as far as can be navigated via right children, starting at the root. The node containing f will be covered in the section for nodes with one child.

In examining the node with value d , only one direction was taken, i.e., no change in direction occurred. But for the other two leaf nodes, there was a change in direction. Further examination yields the following observations regarding the nodes containing e and g :

- There was a change in direction in the path from the root to the leaf.
- The closest values are the parent of the node and the parent of the node where the last change in direction occurred.

If a pointer, initially null, is used to track changes in direction, then once the value to change is found, if it is both an extreme and a leaf, the pointer will have remained null, as no change in direction was ever recorded. In this case, the node's value may be changed in place if the new value is more to the same extreme as the extreme node's parent's data.

For the other cases, additional complexity will be examined to assure all possibilities are being considered. Figure 3 contains a BST with 10 nodes which will permit us to identify the nuances of all the cases for changing data in place.

The three non-extreme leaf nodes each have values between that of their parent and the value in the parent of the node where the most recent change of direction occurred. These are the two locations that must be preserved during the search if it turns out the value to be changed is in a leaf node.

Going forward, the location of the parent of the node to be changed will be denoted *parentOfNode* and the parent of the node where the last change of direction

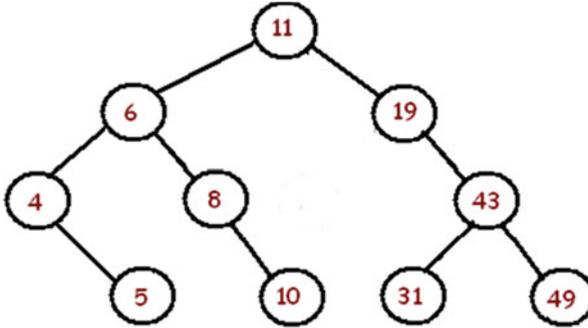


Fig. 3 Another BST

occurred will be denoted *parentOfDirChange*. They should be initially set to null and assigned only when they will correctly reflect the location. This means that the search for the node with the value to change, forthwith denoted *nodeWithData*, will occur with a pointer and a trail pointer.

Assuming the search for the node to change will use pointers named *t* and *trailT*, respectively, the pointers will be assigned as follows:

- *nodeWithData* will be assigned when *t* is equal to the node with the value to be changed.
- *parentOfDirChange* is assigned any time a change of direction occurs.
- *parentOfNode* is assigned the value of *trailT* when the *nodeWithData* is determined.

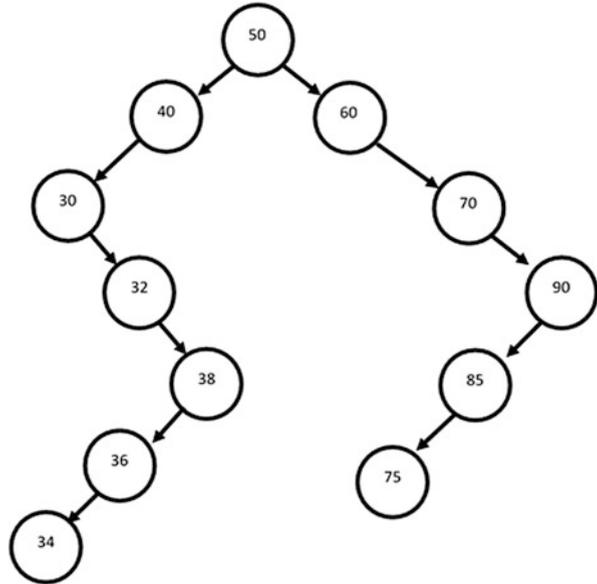
It can be concluded that if a leaf is not an extreme, then its value falls between that of its parent and the parent of the node on the path back to the root where the most recent change of direction occurred. In terms of our pointers, $\text{parentOfDirChange->data} < \text{nodeWithData->data} < \text{parentOfNode->data}$, if *nodeWithData* is a left child and $\text{parentOfDirChange->data} > \text{nodeWithData->data} > \text{parentOfNode->data}$ if *nodeWithData* is a right child.

Node to Be Changed Has One Child

If a node has one child, then there are four possible cases to consider, two of which are unique and two of which are reflective of the other two, respectively. Figure 4 will be used to provide a visual depiction of these cases, which are:

1. *nodeWithData* is a left child that has a left child.
2. *nodeWithData* is a left child that has a right child.
3. *nodeWithData* is a right child that has a left child.
4. *nodeWithData* is a right child that has a right child.

Fig. 4 BST to demo single child cases



These can be boiled down to whether there is a change of direction at nodeWith-Data. Cases 1 and 4 are for no change and 2 and 3 are for where there is a change.

Nodes exemplifying Cases 1 and 4 are {32, 36, 40, 60, 70, 85} while {38} exemplifies Cases 2 and 3. 30 and 90 are not listed, as they are the extreme values in the tree.

The nodes to which Cases 1 and 4 apply have values between that of their parent and their child. Nodes to which Cases 2 and 3 apply will fall between their parent and the parent of the node where the last change in direction occurred.

Pointers already exist to identify these nodes and obtain their data. As for identifying the extreme values in the tree, if parentOfDirChange is null, as well as the child in the same direction, then the current node holds an extreme.

4 Analysis

The worst-case analysis for inserting or deleting a node from a BST is $O(\log_2 n)$. If a node is to be changed, and it can't be done in place or an algorithm isn't implemented to determine whether it can be changed in place, then the BST operations *delete()* and *insert()* will be called, with complexity dependent on the log base two of the size of the data, with actual complexity of twice that. But if the algorithm presented herein is employed, the determination of whether in place editing can occur is also $O(\log_2 n)$, and for a large tree a significant saving can be achieved, as in-place updating is $O(1)$.

Duplicate datum was considered but not applied in developing the algorithm. It does not add complexity to include the possibility of duplicate data; it only requires a set of rules. If it is desired to change a value in a tree that appears multiple times, it must first be established how the multiples are recorded, i.e., whether nodes have a field within which a count can be maintained or each occurrence appears in its own node. For the latter, the algorithm can be applied directly, but if nodes contain a counter for multiples, the user then can be queried whether they wish to update one or all occurrences, and if it is only one occurrence, then the counter can be decremented and the new value inserted. On the other hand, if all occurrences are to be updated, then the in-place algorithm can be applied.

5 Conclusion

The author understands that this algorithm likely doesn't have significant practical use on a BST, but future work will endeavor to examine the same problem on more commonly used data structures, where in-place updating could provide significant gains in efficiency, possibly permitting data to be stored in tree form where previously a high rate of update made that impractical. The motivation for development of this algorithm was educational in nature, as development of this or a similar algorithm is an excellent educational exercise in algorithmic development.

References

1. R.G. Casey, Design of tree structures for efficient querying. *Commun. ACM* **16**(9), 549–556 (1973)
2. P.M. Fenwick, A new data structure for cumulative frequency table. *Softw-Pract. Exp.* **24**(3), 327–336 (1994)
3. W.E. Kluge, Traversing binary tree structures with shift register memories, in *ISCA '76: Proceedings of the 3rd Annual Symposium on Computer Architecture*, (1976), p. 121.1
4. U.-M. O'Reilly, Genetic Programming a Tutorial Introduction, Genetic & Evolutionary Computation Conference (GECCO '11), Power Point Presentation, (2011)
5. R.R. Redziejowski, On arithmetic expressions and trees. *Commun. ACM* **12**(2), 81–84 (1969)
6. K. Shockley, The family binary tree, in *ACM '76: Proceedings of the 1976 Annual Conference*, (1976), pp. 546–550
7. M.A. Weiss, *Data Structures and Problem Solving using Java*, 2nd edn. (Addison-Wesley, 2000), pp. 640–652
8. T. Zaim, Editing a Node in a Binary Tree Structure, (2016), <https://stackoverflow.com/questions/41369799/editing-a-node-in-a-binary-tree-structure>

Class Time of Day: Impact on Academic Performance



Suzanne C. Wagner, Sheryl J. Garippo, and Petter Lovaas

1 Introduction

This paper examines the relationship between the time of day for course scheduling and academic success, controlling for course characteristics and instructor variability. Student academic achievement can be impacted by a variety of uncontrollable factors. The ability for students to schedule classes at specific times throughout the day should enable students to align their preference for morning, afternoon, or evening classes with their predicted academic success in the course. Students who are “morning people” who register for morning classes should perform as well as students who prefer afternoon or evening classes. Ideally, offering the same course at various times throughout the day should allow for the student academic success rates independent of the class time of day.

Prior research has shown variability concerning the impact of the start time of instruction on cognitive performance. A study of high school students indicated that later high school start time led to higher reading test scores for females and that longer sleep led to greater academic success [6]. College students who follow irregular sleep schedules resulting in sleep loss decrease their acquisition and retention of course material [1]. The presence of next-day classes reduced alcohol consumption among college students [2]. Students were found to perform better in the afternoon than in the early morning, suggesting that morning classes hampered student performance [7], and students were found to have earned higher grades in

Type of submission: Regular Research Paper.

S. C. Wagner (✉) · S. J. Garippo · P. Lovaas
Niagara University, Niagara County, NY, USA
e-mail: scwagner@niagara.edu; sgarippo@niagara.edu; plovaas@niagara.edu

classes that start later in the day [3], suggesting that later classes improved student performance. A small positive time of day effect was found to impact student grades [4]; however, morning versus evening classes did not affect test performance [5]. Previous studies show contradictory indications of class time of day on student performance prompting evaluation of a statistics course offered over 13 years to see if the time of day that the course was offered had an impact on student performance.

2 Methodology

The methodology of this study was to compare student grades (as percentages out of 100) for differing class time of day course offerings controlling for course content and course instructor.

The course studied was entitled “Using the Computer as a Research Tool.” The course covers 18 chapters of statistical analyses using the IBM SPSS Statistics software program. The course also covers a short section on report writing using word processing via Microsoft Word and data analysis with graphics via Microsoft Excel to provide students with the skills necessary to write a research paper.

Student grades were compiled from 13 years of data. The data include 38 distinct sections of the course, offered in the fall and spring semesters. Three instructors taught the course; however additional analysis was conducted to determine the effect of the difference in instructors. All three instructors in this research study employed standard handouts and similar designs in all homework assignments and examinations. The number of student grades included in the study was seven hundred and eighty-eight (788).

3 Results

The first hypothesis studied whether there was a significant difference in student grades between two class times: 10:10 a.m. to 11:05 a.m. and 11:15 a.m. to 12:10 p.m. An independent samples *t*-test (see Table 1), using student grades as the dependent variable, found that there is no difference in student grades across these two class time periods (sig. 0.984).

The second hypothesis studied whether there was a significant difference in student grades for the three instructors that participated in the study. A one-way analysis of variance test (see Table 2), using student grades as the dependent variable, found that there is a significant difference in student grades based on instructor (sig. 0.000). Further analysis, using the Bonferroni multiple comparisons Table, found that one instructor’s student grades were significantly different from the other two instructors (sig. 0.000); two of the instructors’ student grades were not significantly different from each other (sig. 1.000). The instructor found to be

Table 1 Independent samples *t*-test

Group statistics		N	Mean	Std. deviation	Std. error mean
Percentage	Time of class 10:10–11:05	257	0.82257	0.130124	0.008117
Grade	11:15–12:10	312	0.82281	0.150423	0.008516

		Independent samples test								
		Levene's test for equality of variances		<i>t</i> -test for equality of means						
		<i>F</i>	Sig.	<i>t</i>	df	Sig. (2-tailed)	Mean difference	Std. error difference	95% confidence interval of the difference	
									Lower	Upper
Percentage	Equal variances assumed	0.949	0.330	-0.020	567	0.984	-0.000240	0.011930	-0.023672	0.023192
Grade	Equal variances not assumed			-0.020	565.627	0.984	-0.000240	0.011765	-0.023347	0.022868

Table 2 ANOVA with Bonferroni multiple comparisons

ANOVA						
Percentage grade						
	Sum of squares	df	Mean square	F	Sig.	
Between groups	1.579	2	0.790	34.104	0.000	
Within groups	18.174	785	0.023			
Total	19.753	787				
Post hoc tests						
Multiple comparisons						
Percentage grade Bonferroni						
(I) Instructor name	(J) Instructor name	Mean difference (I-J)	Std. error	Sig.	95% confidence interval	
					Lower bound	Upper bound
Instructor 1	Instructor 3	0.100925*	0.013302	0.000	0.06901	0.13284
	Instructor 2	0.010541	0.022831	1.000	-0.04423	0.06531
Instructor 3	Instructor 1	-0.100925*	0.013302	0.000	-0.13284	-0.06901
	Instructor 2	-0.090384*	0.020674	0.000	-0.13998	-0.04078
Instructor 2	Instructor 1	-0.010541	0.022831	1.000	-0.06531	0.04423
	Instructor 3	0.090384*	0.020674	0.000	0.04078	0.13998

*The mean difference is significant at the 0.05 level

statistically different from the other two instructors will be referred to as the *unique* instructor resulting in further instructor analysis and control.

The third hypothesis studied whether student grades were significantly different for the following two class times, 10:10 a.m. to 11:05 a.m. and 11:15 a.m. to 12:10 p.m., for those students who took the course from the unique instructor. An independent samples *t*-test (see Table 3), using student grades as the dependent variable, found that there is no difference in student grades across these two class times (sig. 0.132), when only the unique instructor, with grades different from the other two instructors, is considered.

The fourth hypothesis studied whether student grades were significantly different for morning sections of the class versus afternoon and evening sections of the class (all three instructors included). Morning sections were defined with starting class times before 12:00 noon; afternoon and evening sections were defined with starting class times after 12:00 noon. An independent samples *t*-test (see Table 4), using student grades as the dependent variable, found that there is a significant difference in student grades across the two groups (morning sections and afternoon sections) (sig. 0.001). Students in the morning sections earned significantly higher grades (mean of 82.3%) than students in the afternoon and evening sections (mean of 77.4%).

The fifth hypothesis studied whether student grades were significantly different for morning sections of the class versus afternoon and evening sections of the class

Table 3 Independent samples *t*-test

Group statistics					
Time of class	<i>N</i>	Mean	Std. deviation	Std. error mean	
10:10-11:05	150	0.78280	0.142886	0.011667	
11:15-12:10	224	0.80647	0.152223	0.010171	
Independent samples test					
	Levene's test for equality of variances		<i>t</i> -test for equality of means		
	<i>F</i>	Sig.	<i>t</i>	df	Sig. (2-tailed)
Equal variances assumed	0.147	0.702	-0.1.510	372	0.132
Equal variances not assumed			-1.530	333.023	0.127
				Mean difference	Std. error difference
				-0.023673	0.015673
				-0.023673	0.015478
				Lower	Upper
				-0.054492	0.007145
				-0.054119	0.006773

95% confidence interval of the difference

Table 4 Independent samples *t*-test

Group statistics		<i>N</i>	Mean	Std. deviation	Std. error mean					
Percentage	a.m. versus p.m. a.m. sections	568	0.82272	0.141618	0.005942					
Grade	p.m. sections	123	0.77380	0.160703	0.014490					
Independent samples test										
		Levene's test for equality of variances		<i>t</i> -test for equality of means						
		<i>F</i>	Sig.	<i>t</i>	df	Sig. (2-tailed)	Mean difference	Std. error difference	95% confidence interval of the difference	
Percentage	Equal variances assumed	3.348	0.068	3.388	689	0.001	0.048917	0.014438	Lower	Upper
Grade	Equal variances not assumed			3.123	165.477	0.002	0.048917	0.015661	0.020568	0.077266
									0.017996	0.079838

for those students who took the course from the unique instructor. An independent samples *t*-test (see Table 5), using student grades as the dependent variable, found that there is a significant difference in student grades across the two groups (morning sections and afternoon sections) (sig. 0.001), when only the unique instructor's grades were considered. Students in the morning sections earned significantly higher grades (mean of 79.7%) than students in the afternoon and evening sections (mean of 73.6%).

The sixth hypothesis studied whether student grades were equal for morning sections of the class versus afternoon sections of the class for those students who took the course from the two instructors with statistically similar grades (i.e., not including the unique instructor). An independent samples *t*-test (see Table 6), using student grades as the dependent variable, found that there is no significant difference in student grades across the two groups (morning sections and afternoon/evening sections) (sig. 0.717), when only the two instructors' grades were included.

4 Conclusions

There is no significant difference in student grades among the morning sections of course offerings with class starting times of 10:10 a.m. versus 11:15 a.m. Although registration data indicates that students prefer the 11:15 a.m. class over the 10:10 a.m. class, the results indicate no significant difference in performance between the two morning classes. When considering course offerings in the morning vs. the afternoon/evening sections, the results indicate that, when all three instructors are included, student grades are significantly higher for those students in the morning sections of the class versus those in afternoon and evening sections of the class. However, the unique instructor may impact the performance difference in morning versus afternoon/evening classes since no significant difference was found for the courses taught by the two other instructors.

The results of analyzing the difference in instructors revealed a significant difference in student grades among instructors, more specifically between the unique instructor and the other two instructors. The unique instructor that participated in the study had significantly different student grade percentages than the other two instructors. Although all three instructors used the same class format including design of homework assignments and examinations, the individual instruction impacted student performance as measured by student grades. When only the two instructors with similar grades are included, there is no significant difference in student grades for those students in the 10:10 a.m. versus 11:15 a.m. sections and no significant difference in student grades for those students in the morning sections of the class versus those in afternoon and evening sections of the class. The results of this study show that student performance is not impacted by the time of day that a course is offered.

Table 5 Independent samples *t*-test

Group statistics		<i>N</i>	Mean	Std. Deviation	Std. Error Mean
Percentage	a.m. versus p.m. a.m. sections	374	0.79698	0.148808	0.007695
Grade	p.m. sections	87	0.73598	0.168533	0.018069

		Independent samples test								
		Levene's test for equality of variances		<i>t</i> -test for equality of means						
		<i>F</i>	Sig.	<i>t</i>	df	Sig. (2-tailed)	Mean difference	Std. error difference	95% confidence interval of the difference	
									Lower	Upper
Percentage	Equal variances assumed	1.823	0.178	3.356	459	0.001	0.061002	0.018176	0.025284	0.096719
grade	Equal variances not assumed			3.106	119.118	0.002	0.061002	0.019639	0.022115	0.099888

Table 6 Independent samples *t*-test

Group statistics		<i>N</i>	Mean	Std. deviation	Std. error mean
Percentage	a.m. versus p.m.	194	0.87235	0.111192	0.007983
Grade	a.m. sections	36	0.86522	0.089865	0.014987
	p.m. sections				

		Independent samples test	
		<i>t</i> -test for equality of means	
		Levene's test for equality of variances	
		<i>F</i>	Sig.
Percentage grade	Equal variances assumed	0.110	0.741
	Equal variances not assumed	0.420	0.518
		<i>t</i>	df
		0.363	228
		0.420	56.879
		Sig. (2-tailed)	
		0.717	0.676
		Mean difference	
		0.007128	0.007128
		Std. error difference	
		0.019634	0.016972
		95% confidence interval of the difference	
		Lower	Upper
		-0.031559	0.045815
		-0.026860	0.041116

References

1. M. Baynard, D. Mceachron, Work in progress – 2014; Asleep in class are the schedules of college students hampering their ability to learn? 2011 Front. Educ. Conf. (FIE) (2011). <https://doi.org/10.1109/fie.2011.6142953>
2. H.L. Berman, M.P. Martinetti, The effects of next-day class characteristics on alcohol demand in college students. *Psychol. Addict. Behav.* **31**(4), 488–496 (2017). <https://doi.org/10.1037/adb0000275>
3. C. Cotti, J. Gordanier, O. Ozturk, Class meeting frequency, start times, and academic performance. *Econ. Educ. Rev.* **62**, 12–15 (2018). <https://doi.org/10.1016/j.econedurev.2017.10.010>
4. A.K. Dills, R. Hernández-Julián, Course scheduling and academic performance. *Econ. Educ. Rev.* **27**(6), 646–654 (2008). <https://doi.org/10.1016/j.econedurev.2007.08.001>
5. C. Gao, T. Terlizzese, M.K. Scullin, Short sleep and late bedtimes are detrimental to educational learning and knowledge transfer: An investigation of individual differences in susceptibility. *Chronobiol. Int.* **36**(3), 307–318 (2018). <https://doi.org/10.1080/07420528.2018.1539401>
6. J.A. Groen, S.W. Pabilonia, Snooze or lose: High school start times and academic achievement. *SSRN Electron J.* (2018). <https://doi.org/10.2139/ssrn.3280312>
7. K.M. Williams, T.M. Shapiro, Academic achievement across the day: Evidence from randomized class schedules. *Econ. Educ. Rev.* **67**, 158–170 (2018). <https://doi.org/10.1016/j.econedurev.2018.10.007>

A Framework for Computerization of Punjab Technical Education System for Financial Assistance to Underrepresented Students



Harinder Pal Singh  and Harpreet Singh

1 Introduction

Indian higher education system is largely based on University Grants Commission (UGC) guidelines for universities, which have a large number of affiliated colleges across the country. UGC also act as a regulator for general higher education in the country [1]. Similarly All India Council for Technical Education (AICTE) guides large number of engineering colleges, polytechnics, technical universities, and their affiliated colleges and act as a regulator for engineering and other professional education in the country. The certificate level skill/vocational courses are largely run by Industrial Training Institutes (ITI) under the guidelines of Director General of Employment and Training (DGET) and are regulated by the National Council for Vocational Training (NCVT). The directorates of technical educations (DTEs) are state government organizations that are managing the technical education system at state levels under the guidelines of the above referred federal statutory bodies like UGC, AICTE, DGET, and NCVT. There are tens of thousands of colleges affiliated to universities and state boards of technical education, and millions of students are searching for courses of their choice and financial assistance to fund their education. There are many federal and state-funded financial assistance schemes available for students of different economic status in the society like for weaker sections of the society-scheduled castes (SC) and other backward classes (OBC), minority communities, and merit cum means scholarships are available. The univer-

H. P. Singh (✉)

Department of Electronics and Communication Engineering, Desh Bhagat University and
Department of Technical Education and Industrial Training, Mandi Gobindgarh, Punjab, India

H. Singh

Department of Electrical and Computer Engineering, Wayne State University, Detroit, MI, USA
e-mail: hsingh1@wayne.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_24

337

sities, colleges, polytechnics, and industrial training institutes look for solution to handle such large amount of student data ranging from enrollments, scholarships, academic content, literature search, testing, results, and finally placements and also for analyzing data for future decision-making. Such educational institutions are generating huge volumes of data, from grades or test scores to enrolment numbers, scholarships, and placement tracking. With the advent of online courses offered by many universities, the amount of data available to educational officials and students has exploded [2]. Various database management frameworks and analytical software help in identifying relevant pedagogic approaches. The new frameworks are needed in today's world to support data mining approaches for increasing the efficacy of educational institutions. For providing financial assistance to unrepresented students, a new framework for computerization of Punjab Technical Education has been suggested which is more secure, cost-effective, reliable, and easy to use and can handle ever-increasing memory space requirements. Modern-day open-source tools like MongoDB coupled with cloud technology are used, thereby making comparisons with other conventional technologies. The results are interesting as summarized in the later sections. Some characteristics of NoSQL databases are inherently schema-less and highly scalable. Also due to advances in the information and communication technology and faster Internet facilities, it is much easier for institutions like schools, colleges, and universities to approach out to more and more students and to attract them for admissions, academics, scholarships, and other similar and related activities. Such data generated in technical and engineering education system may be further classified such as data related to the financial assistance/scholarships, admissions, academics, and evaluation. Universities and other educational institutions are working overnight to identify relevant talent pools and new courses with a view to appeal to the students based upon such data analysis. Scholarships and financial assistance options are available for weaker sections of the society and other minority communities through federal and state funding. As per the 2011 census of India, for a total of 1.2 billion of population, the scheduled castes and tribes (SC/ST), backward classes, and minority community were having major numbers. As shown in pie chart in Fig. 1, the scheduled caste population was about 19.59% and scheduled tribes (ST) was 8.63% of the total population in India. Other backward classes (OBC) population was 40.94%. Similarly population of minority community in India was about 20.5% which is a considerable chunk as shown in Fig. 2. Most of the scholarships and financial assistance schemes were designed for uplifting such weaker and unrepresented sections of the society and also uplifting the minority communities with low incomes. Millions are being spent for funding education of such students year after year in the past many years throughout the country with spending ratio of federal-states 90:10. The bar chart in Fig. 3 shows students applying for financial assistance/scholarship schemes only for the state of Punjab having about 1800 affiliated colleges and polytechnics and Industrial Training Institutes. Numbers are much higher at country level. Annual spending on such social justice schemes is more than INR 60000000 in the state of Punjab alone. India consists of about 30 such states, and similar social welfare schemes are running in all the states, covering large number of students. When

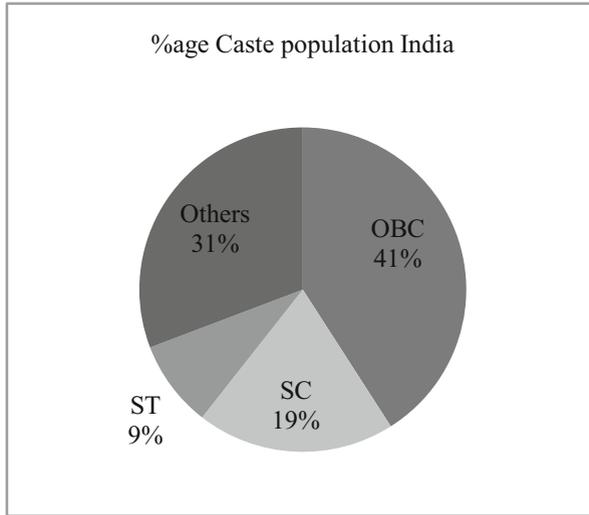
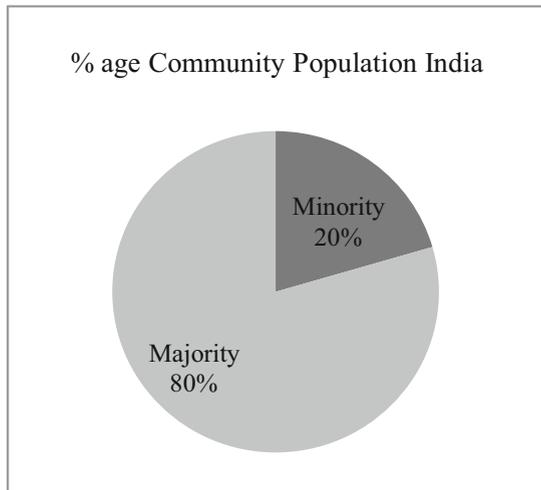


Fig. 1 Caste-based population, 2011

Fig. 2 Community-based population, 2011



such a large no. of students apply for scholarships and financial assistance, it is very difficult to eliminate the students with duplicate data, fake data, suspicious and unreliable data arriving from so many sources year after year, and every new intake. Global education systems may already be using advanced tools and using such advanced practices for real business intelligence, financial analytics, and predictive analytics and finally making the strategic management to remain effective. The data sources may include students' personal information, their results, certificates, past educational qualifications and institutions, and parental income and dropout rates,

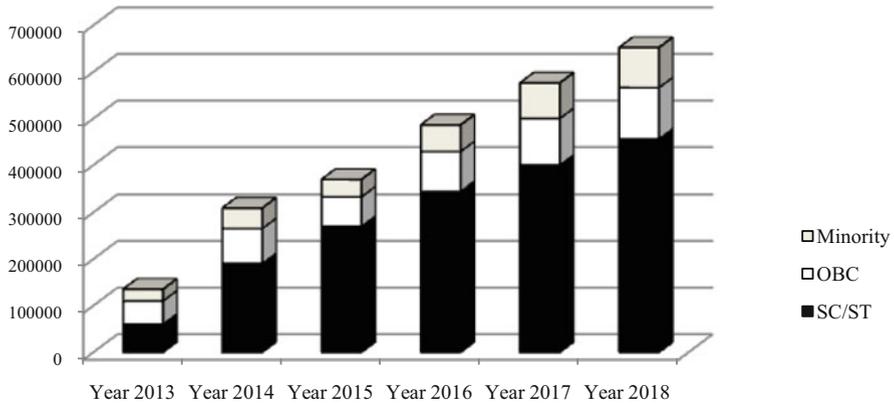


Fig. 3 Students applying for financial assistance year wise in the State of Punjab

including some sensitive information like their social security number/UID (unique identification) number, bank account number, etc.

2 Financial Assistance Management

2.1 Risk Detection

Data security and information integrity is a big challenge in institutional data as the personal data and information of applicants can be stolen online. For example, if national identity number (UID) of the student or bank accounts are stolen by hackers, it can lead to financial loss to the applicant students. Leakage of such personal and classified data can lead to various scams. So risk detection and analysis and using various security techniques like modern encryption algorithms are proposed to be inbuilt in the data mining system [3].

2.2 Performance Prediction

The performance prediction of students whether he/she is continuing in his studies after availing the benefit of financial assistance need to be ascertained before granting the scholarship application for the next semester/year. His/her board/university scores need to be linked using various data tools to the database management system. If he/she does not appear or pass any of the subjects, his/her application is liable to be rejected till he/she passes the requisite number of subjects and reapply for scholarship of next semester/year. In the proposed study data, alert has been

implemented. Dropout rates can be ascertained while analyzing the data, so finally the decision-making can be improved for further award of scholarships.

2.3 Data Visualization

Technical educational data become more and more complex as it grow in size. Data can be visualized using data visualization techniques to easily identify the trends and relations in the data just by looking on the visual reports.

2.4 Intelligent Feedback

Learning systems can provide intelligent and immediate feedback to students in response to their inputs which will improve student interaction and performance. It is proposed to implement a new framework that can be developed by linking application submission transaction for scholarship applications till the approval happens.

2.5 Conventional Database Framework

There are tens of thousands of students applying for different financial assistance and scholarships every intake. There are number of options and schemes available on the basis of caste, social status of families, merit cum means, or uplifting of minority communities. The step-by-step procedure is shown in Fig. 4 which is currently in place.

2.6 Implementation of New Framework

The following Ford-Fulkerson algorithm for new framework describes it as:

```
input: Applications form from students
output: send checks  $f$  to banks for awards
for each application  $(u, v)$  in Database do
implement clustering approach to distribute the applications
while there exists appropriate application to scholarship.
return  $f$ 
```

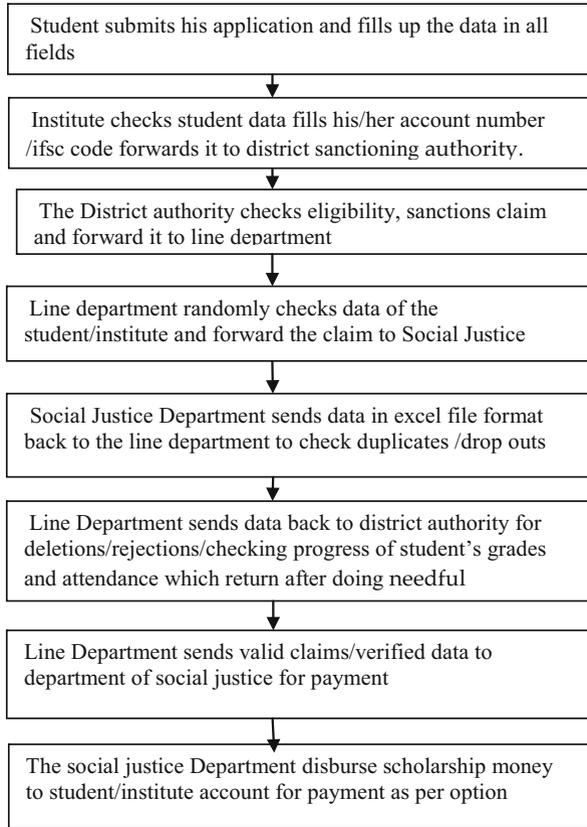


Fig. 4 Flow chart of existing framework

It is based on the following example:

Here follows a longer example of mathematical-style pseudo-code, for the Ford-Fulkerson algorithm:

```

    Algorithm Ford-Fulkerson is
    input: Graph  $G$  with flow capacity  $c$ ,
           source node  $s$ ,
           sink node  $t$ 
    output: Flow  $f$  such that  $f$  is maximal from  $s$  to  $t$ 
    (Note that  $f_{(u,v)}$  is the flow from node  $u$  to node  $v$ , and  $c_{(u,v)}$ 
    is the flow capacity from node  $u$  to node  $v$ )
    for each edge  $(u, v)$  in  $G_E$  do
     $f_{(u, v)} \leftarrow 0$ 
     $f_{(v, u)} \leftarrow 0$ 
    while there exists a path  $p$  from  $s$  to  $t$ 
    in the residual network  $G_f$  do
    let  $c_f$  be the flow capacity of the residual
    network  $G_f$ 
     $c_f(p) \leftarrow \min\{c_f(u, v) \mid (u, v) \text{ in } p\}$ 
  
```

```

    for each edge (u, v) in p do
    f(u, v) ← f(u, v) + c_f(p)
    f(v, u) ← -f(u, v)
    
```

2.7 Description of Framework

The new framework is cloud-based platform using virtualized cluster of servers over data centers over SLA [4]. Dynamic resource provisioning of the servers storage and the networks is cloud computing basically. The student fills in the application details from his/her mobile phone/laptop. The UID server authenticates his/her identity from his/her UID (unique identification) number and opens up the application form. The student fills it up, attaches and uploads eligibility documents, and submits it online. College/university server checks his/her academic, enrollment, and performance credentials and forwards his/her application online to district sanctioning authority. District sanctioning authority ascertains the eligibility documents and sanctions the student claim which goes to line department. The line department collects all claims; checks authenticity of district sanctioning authority, university/-college affiliation and recognition status, and the upper limit of amount claimed;

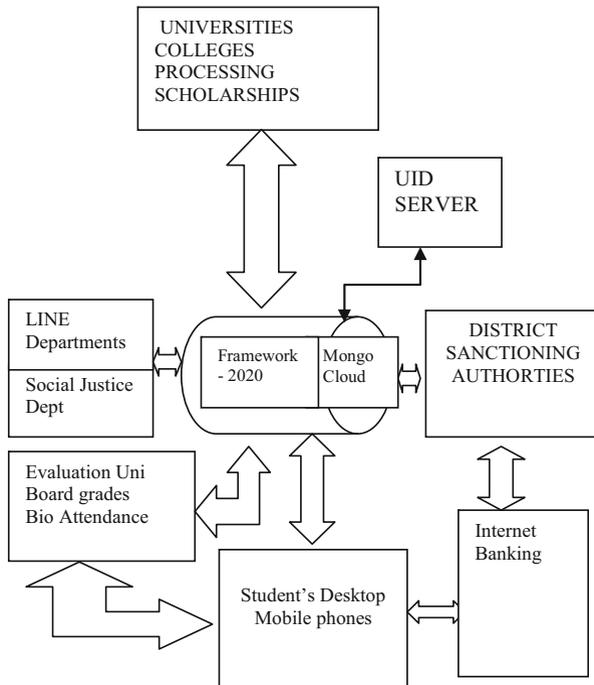


Fig. 5 New framework of computerization of Punjab Technical Education

and finally checks the attendance performance from the linked university/board server and sends the claims for releasing payment to the Department of Social Justice. The block diagram of new framework is shown in Fig. 5. The Department of Social Justice sends the money to UID linked bank account of the student through Internet banking as the account number and IFS code data in the student application. Many students are doing multiple times same activities year after year till they pass. This kind of big data generated ranging from admission, claiming financial assistance, attendance, performance, etc. is stored in mongo cloud and available for decision-making and analysis by the line department and the Department of Social Justice for arranging funds, estimations, budgeting, and other decision-making analytics. MongoDB data base architecture in the new framework is more secure than MySQL based old data base model where lot of server memory and some manual processing of re-verifying the performance of student was required also manual deleting fake/duplicate claims and there were delays in releasing the scholarships and financial assistance to the students account [5].

2.8 Hardware and Software Specifications

The new data model require only a server and the application software giving access to mongo cloud platform, which can be hired for need-based memory requirements. In the present case, existing national informatics (NIC) server is sufficient for controlling the activity. The NIC server hosts the software application controls. There is no need for adding more hard disk memory or other memory which may become expensive year after year. The application software shall be connecting all the existing servers like mongo cloud, university/board server for student performance query, Internet banking, and (unique identification authority) UID server to student mobile phones, laptops, or tablets. For this software application, the student can install on his/her mobile phone or use laptop to access the application from the Internet using normal browser. All the data can be added and processed simultaneously using the application interface. The hardware parallel processing diagram is given as simple illustration in Fig. 6.

In the following section for developing a computer and mobile application, the basic algorithms for importing existing data to MongoDB are given.

3 Importing Data to MongoDB and Comparison

3.1 Importing CSV File into MongoDB

Create a folder on disk C, c:\importMongo, then download the file “Import-DataMongo.rar” from Google Drive, then extract file ImportDataMongo.exe from

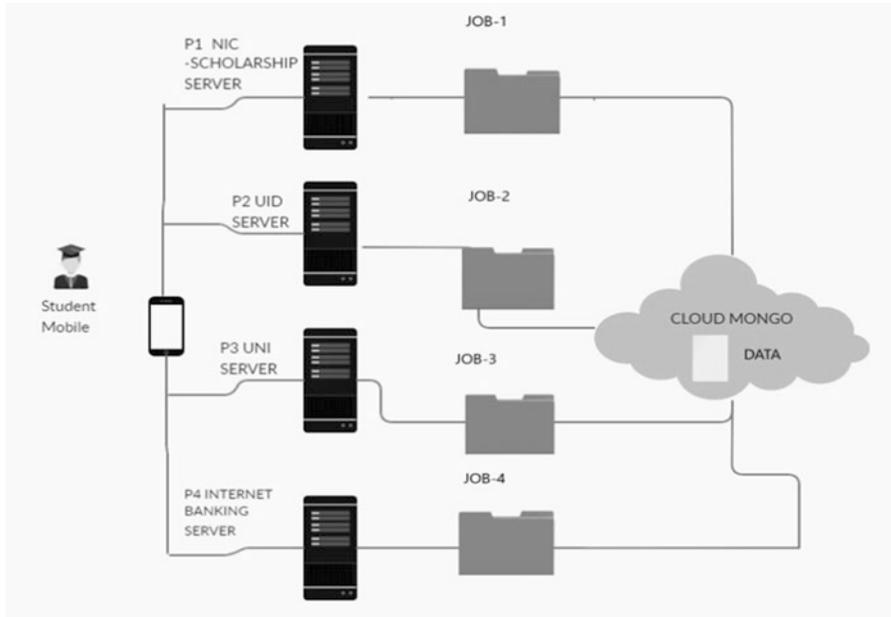


Fig. 6 Parallel processing layout depicting jobs of all four servers connected to cloud

“ImportDataMongo.rar” in the folder c:\importMongo, then copy here the CSV file which you want to import to Mongo in the folder c:\importMongo, then launch command prompt, and change folder to c:\importMongo.Run the file c:\importMongo\ImportDataMongo.exe

Note: the CSV file after import will be moved to the folder c:\importMongo\Archive

3.2 *Checking the Imported Data in MongoDB*

Launch the application Compass from MongoDB. Click on Sample_StdRec, click on stdRecords, and click on Table.

3.3 *Comparison of New Framework with Other Database Systems*

The field of education is gaining insight from large volumes and variety of real-time student data. Educational institutions are generating huge volumes of data, from grades or test scores to enrolment numbers and scholarships [6]. The big

data paradigms are needed in today's world to support data mining approaches for increasing the efficacy of educational institutions [7]. The usage of MongoDB platform for data storage and analyzing educational data is proposed. Modern-day open-source tools like MongoDB coupled with cloud technology are being used to test real data samples on a real-time basis for analysis of students' scholarships data using graphs and charts of the realistic data, and comparisons thereof with other conventional technologies are carried out. These databases support frameworks like MapReduce for processing of large amounts of data in parallel fashion. The MapReduce framework deals with data mapped on distributed file systems, with intermediate data being stored on local disks and can be retrieved remotely by reducers [8]. Google's proprietary MapReduce paradigm reads and writes to the Google File System, i.e., GFS. But recently certain platforms like MongoDB, Apache Hadoop HDFS, Hive, Bigtable, HBASE, etc. have emerged to store large amounts of data. MongoDB is useful for storing educational data as this is NoSQL, open-source document-oriented database system developed by 10Gen company. MongoDB stores structured data as JSON-like heterogeneous documents with dynamic schemas, and it scales horizontally. It also has a functionality of querying database and suitable for storing educational data due to its scalability and flexibility in structural format for storage. The platform is useful for content management and delivery and is attractive due to features listed below:

- (a) Data are stored in the form of JSON style documents and uses simplified JavaScript engine.
- (b) It supports GridFS for storing data.
- (c) MongoDB is a document database in which one collection (i.e., data store) can hold a variety of documents. Number of fields, content, and size of the document can be different from one document to another.
- (d) Conversion of application objects to structural format of database objects not needed.
- (e) No complex joins, as in traditional database systems.
- (f) MongoDB supports dynamic queries on documents using a document-based query language.
- (g) MongoDB is easy to scale.
- (h) Uses internal memory for storing the (windowed) working set, enabling faster access of data.
- (i) Index on any attribute could be made and fast in-place updates on data.
- (j) It supports replication, shredding, and high availability.

4 Results and Discussion

Punjab Technical Education is providing financial assistance to underrepresented students for post-matric scholarship (PMS) scheme. This scheme enables free education for scheduled caste (SC) students and other backward class (OBC)

Table 1 Year-wise SC students and claims in INR

Year	Numbers	Claimed amount	Dropout/duplicate numbers
2014–2015	57440	2489978482/–	8283
2015–2016	54276	2608049719/–	3538
2016–2017	64029	3108739314/–	6342
2017–2018	56825	2846101493/–	7698
2018–2019	38885	2005609086/–	3207

Table 2 Year-wise OBC students and claims in INR

Year	Numbers	Claimed amount	Dropout/duplicate numbers
2014–2015	9854	477288913	1373
2015–2016	9153	447747836	451
2016–2017	2448	121241411	289
2017–2018	952	45566433	110
2018–2019	435	21513161	37

students, whose parent's annual income is less than INR 250,000 and INR 100,000, respectively, and their minimum education in each case 10th standard high school. As per the schedule of the Department of Social Justice and Empowerment of Minorities, which is an implementing department, the students can apply for financial assistance every year. The payment is made directly to the UID linked bank accounts of students/institutes by the Department of Social Justice and Empowerment of Minorities, Punjab. The data for eligible students is processed by Punjab Technical Education department (DTE) which is designated as one of the line department. Other line departments are Department of Medical Education, Department of Higher Education, and Department of School Education.

The following consolidated data table shows year-wise financial assistance claimed by underrepresented students belonging to scheduled caste (SC) whose parents income is less than INR 250000 and other backward classes (OBC) whose parents income is less than INR 100000. The data shown is for Punjab Technical Education (DTE). Discussion on the data shows a lot of money is being disbursed to the students, and there are considerable no. of students applying for such financial assistance every year. Similarly students of other line departments like medical education, higher education, and school education are also applying for the same as all the students seeking any kind of education are eligible to apply if they satisfy general eligible conditions. There is a risk of duplicate claims as same student may be applying with other line departments as the data shows up in the last column of table. Manual and conventional data management frameworks were not fruitful as financial implications were involved (Tables 1 and 2).

Also there were data security risks as UID numbers and bank accounts were part of data of personal information of the students.

5 Conclusion

It is successfully concluded that the new framework for computerization of financial assistance to unrepresented students will help in weeding out duplicity of claims with other scholarship schemes of the State of Punjab Technical Education and similar schemes of the country as UID server authenticates the student ID before application form opens up. It will also help in asserting the student performance like checking attendance and grades from university server simultaneously without any delays whatsoever. It also helps in transparency in processing the student claims as there is no manual interface with students and authorities. It also helps the Punjab Technical Education with increased data security and authorized access of data due to capabilities of using MongoDB-based tool and saving lots of hardware memory space as the cloud technology is used and need-based cloud server can be hired. Considerable improvements in data query times can also be achieved. This may save them cost and time apart from avoiding possible frauds and scams. In the future, students may use only smart phones for all kinds of educational activity so this framework will come handy for them. Also in the future, the usage of combinations of various platforms like Hadoop, MongoDB, Cassandra, etc. and parallel programming models like Hadoop, MapReduce, PACT, etc. for various data analytics techniques could be explored to accelerate the analysis of educational data. This will help in building scalable models in the field of education and may provide a better scope of improvement in the field of educational analytics as unstructured data from social media networks can also be utilized to know the student interests.

References

1. Dhirendra Pal Singh- Chairman UGC- 2019, Annual Report 2018–19 of University Grants Commission, India, Available at: https://www.ugc.ac.in/pdfnews/3060779_UGC-ANNUAL-REPORT%2D%2DENGLISH%2D%2D2018-19.pdf
2. B. Daniel, University of Otago, New Zealand, Big data and analytics in higher education: Opportunities and challenges. Br. J. Educ. Technol.: BJET (2014). <https://doi.org/10.1111/bjet.12230>
3. Y.H. Kim, J.-H.A. Hoseo, A Study on the Application of Big Data to the Korean College Education System, in *Conference paper (Information Technology and Quantitative Management (ITQM 2016) University of Korea)*,
4. K. Hawang, G.C. Fox, J.J. Dongarra, Distributed and cloud computing from parallel processing to the internet of things. **11**(5), 35–48. publication Morgan Kaufmann, an imprint of Elsevier 225 Wyman Street, Waltham, MA 02451, USA *JCSI* (2014)
5. H.P. Singh, H. Singh, Big data technologies in scholarship management of technical education system. Int. J. Emerg. Technol. Innov. Res. (www.jetir.org | UGC and issn Approved), ISSN:2349–5162 **6**(6), 492–496 (2019) Available at: <http://www.jetir.org/papers/JETIR1906H50.pdf>

6. A.S. Drigas, P. Leliopoulos, The use of big data in education. *Int. J. Comp. Sci. Issues (I)*
7. A. Cuzzocrea, Warehousing and Protecting Big Data: State-Of-The-Art-Analysis, Methodologies, Future Challenges, in *ICC '16: Proceedings of the International Conference on Internet of Things and Cloud Computing.*, Article Number 14, (2016). <https://doi.org/10.1145/2896387.2900335>
8. M. Darrell, *Big Data for Education: Data Mining, Data Analytics, and Web Dashboards* (West, 2012)

Parent-Teacher Portal (PTP): A Communication Tool



Mudasser F. Wyne, Matthew Hunter, Joshua Moran, and Babita Patil

1 Introduction

In this paper we discuss the importance of research in maintaining good school-home partnership through parent-teacher relationship. In [1], the authors conducted a review of two studies analyzing risk factors of outcomes of student's lack of involvement. The paper details the importance of parent involvement in relation to the outcomes of students' correlated success. This study analyzed three areas of focus with regard to a parent's involvement in their child's academic studies. These categories were broken down into two main categories, the parent-child involvement and the parent-school involvement. Parent-child involvement measured three categories that measured how much communication the parent and child had when discussing homework and general school discussion. The parent-school involvement measured how much communication was occurring and at what intervals parents were actively engaging on behalf of their child [1]. We were curious of the results in the findings in Finn's analysis in [1] and continued to conduct additional research. We reason that if parent involvement was a factor that was strongly considered in his analysis, there must be a body of evidence that has correlated in the past. The authors in [2] state that it is difficult to base findings of academic success and correlating factors because there is a large body of variables that are nearly impossible to factor. The authors reiterate that there is a large body of evidence to suggest parent involvement is beneficial to a child's success and teachers and parents need to partner together in helping children reach their full potential.

High school teachers are not completely opposed to the idea of communication with parents of their students. They think that this communication, via email or

M. F. Wyne (✉) · M. Hunter

School of Professional Studies, National University, San Diego, USA

e-mail: mwyne@nu.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_25

351

phone, should only be in case of emergency and not on routine basis. On the other hand, we also acknowledge and understand that some teachers may not communicate at all. Research shows that children do better in school when parents talk often with teachers and become involved in the school. There are number of ways that parents and teachers can communicate with each other, rather than relying on the scheduled parent-teacher conferences. Close communications between parents and teachers can help the student. In addition, parents who participate in school activities and events will have added opportunities to communicate with teachers. Becoming involved with parent-teacher organizations gives the teacher and parent the possibility to interact outside the classroom. This will also present an opportunity for the parent to provide input into decisions that may affect their child's education. Sometimes teachers conduct welcome meeting with their students' parents early in the school year, in an effort to better understand parents and their children and how to support their education. Teachers appreciate knowing that parents are concerned and interested in their child's progress; this also helps open the lines of communication between teachers and parents.

Parent-teacher conferences are often scheduled at the time of the first report card for the school year. For parents and teachers, this is a chance to talk one-on-one about the student. This conference is a good occasion to launch a partnership between parent and teacher that will function during the school year. A good investment in one's children education is to volunteer, depending upon parent's availability, interests, skills, and the needs of the school; the opportunities are endless, at times school personnel may not know what parents want to do as a volunteer. Some suggestions include lunchroom monitor, tutoring, library aid, classroom speaker on a specific topic of interest, and concession worker at school events. Phone calls and visits to the classroom are also good ways to discuss appropriate times and means of contact with the teacher to open communication channel between parents and teachers and stay informed about child's progress.

The study conducted by the National Household Education Surveys Program reports that 59% of parents whose children are going to public schools complained that they never had any communication from teaching staff [3]. Lack of communication from teachers of the public school can also be attributed to the fact that each teacher is responsible for a large number of students, so they do not have time for one-to-one communication with parents. Some of them have also reported that they have, at time, outdated phone number and/or email address, thus making it impossible to establish any kind of communication.

Schools need to require teacher communication with their student's parents and not an optional responsibility. This will result in teacher either making time during school day or spending additional time. In addition, teacher will also need some kind of guidance and training on the form of communication as well as contents of the messages to be sent. One must keep in mind that low-income school with underachieving students will present additional challenges. During parent-teacher or back-to-school meetings is an excellent opportunity of parents to talk to their children's teachers but only a few parents generally show up for such meetings, especially not the ones that you really need to talk to.

Texting can be considered as for the most underused and promising form of parent-teacher communication [3]. These days, phones, even for poor families, can be considered as a convenient and inexpensive mode of communication for pushing vital news as well as allowing teachers to reach out to parents with a personalized message. These messages can be very short and sweet to remind parents of the upcoming tests, assignments, or even future school trips and meetings. With continued and clear communication from the start of the academic year by the teacher, it can avoid any misunderstanding on the parents' side as well as will keep them informed on class, school, and community activities. Such approach may lead to having very understanding and supportive parents [4]. The authors in [5] suggest that communication should be initiated with parents as soon as the list of students in the class becomes available; it would be very helpful especially for teachers who are new in the field. The communication may include a brief self-introduction and contact information. In addition, the authors in [6] present factors that affect the development of effective parent-teacher relationship. These factors include matching between parent-teacher culture, societal forces for school and family, and how parents as well as teachers view their individual roles.

The work reported in this paper is an attempt to fill the communication gap between teachers and parents. At the heart of this issue is a lack of involvement or ability to communicate using traditional methods. By developing a web-based portal, we are attempting to provide tools and methods that are in use and utilized by millions of users. We hope to use tools such as a web portal, email, text, calendars, and event reminders to assist teachers and parents to engage in their students learning experience.

2 Parent-Teacher Portal

Good two-way communication between parents and teachers is necessary for student's success. Research shows that the more parents and teachers share relevant information with each other about a student, the better equipped both will be to help student to succeed academically. In addition, an effective parent-teacher communication also helps children do better socially. This is especially important for a child who is struggling in school, as seeing his parents and teachers working together to solve the problem can be tremendously reassuring. When children are in elementary school where they are just starting the school or transitioning between one grade to another, it becomes even more important for the parent and teacher to be knowledgeable about the student's progress. Although teachers and parents recognize the importance of effective parent-teacher communication, few gleefully anticipate the actual occasions of that communication. Many teachers, while fully aware of the importance of effective parent-teacher communication, still dread the actual occasions of that communication. It is not easy to maintain or promote home-school partnership since it also depends on size of the community where the school is located; in smaller communities, it is much easier because of intimate

connections. In any case, it is not easy to either initiate or maintain communication with the parents; especially with difficult parents, it can be very stressful. In order to support communication between teacher and parents, [7] presents four tools such as *Remind*, *ClassDojo*, *Bloomz*, and *ClassTag*. Comparison of these tools with our PTP will be presented later in this paper.

To help make parent-teacher communication easier, clear, and precise, having an app that both the parent and the teacher can use with ease would be of great benefit; hence we decided to develop an online parent-teacher communication app named *Parent-Teacher Portal (PTP)*. The PTP app can help bridge communication gaps between the parent and teacher. The proposed portal is based on the following:

- What do teachers and parents need to talk about?
- How can learning experiences be designed that require the parent-teacher interaction?
- How can we make app a reliable and effective tool in child's development?
- Will using an app make it easier for the parent and teacher to initiate and have a continuous communication?

For Teachers The app makes it easier for teachers to send out assignments, reminders, and progress reports as well as communicate with parents about conferences, field trips, volunteer opportunities, and school material donations. Teachers can send messages to individual parents, to a group, or to the whole class. They can also attach pictures to messages or conduct polls to ask for parent feedback. Teachers can also choose to share classroom events and photos, giving families a chance to feel more connected to the classroom.

For Parents The app provides parents a reliable medium to contact teachers, reply to the teacher's message, and ask questions pertaining to their child. Class group chats can be created where parents can submit generic questions, suggestions, etc. Parents can also participate in one of the existing group chats.

3 Existing Tools

In this section we are highlighting important functionality potential and a comparison to our proposed PTP.

3.1 *Remind*

Remind provides some of the features that we have but more centered on messaging. This messaging would provide translation services, text messaging, read receipts, and provide quick references to what was covered during the school day. There are additional services, but they would require an upgrade for additional cost.

We were unable to receive pricing information for these services. To receive this, one would need to continue through with the registration/upgrade process. These additional services include broadcast messaging and longer messages (presumably 255 characters is the max.), rostering, admin controls, and statistical analysis. There is also a limit on the number of classes that can be added per account. The class size is limited to 10 per account. However, it is unclear if that number is for each teacher or each school. If that limit was on teachers, 10 is an acceptable size limit as most teachers in elementary school will only have 1–2 classes a year. The biggest issue here is that there would be no ability to achieve this information if it would be needed at a later date. If the limit is opposed on a school level account, this would not adequately address the demand and only serve individual teachers who chose to use the *Remind* app. Using a quick search on the support page of *Remind* site, the very first support entry is titled “My district is banning *Remind* this year. What can I do?” For this reason, we envisioned the PTP being more of an integral part of a school/district to avoid issues of privacy, support, or law. For this reason, the application would first need to be deployed in a more regional setting. As a national enterprise, there would be a strong argument to not allow a third party the ability/opportunity to perform data analytics on children. This application would be a direct competitor to our PTP application and would provide similar services when compared to our functional requirements in communication. As far as we could tell, *Remind* does not provide a calendar service.

3.2 *ClassDojo*

ClassDojo provides a system in which teachers are actively engaging students and providing real-time updates to the student’s parents. *ClassDojo* provides a means to integrate communication with curriculum and provides students a fun interactive way to participate within that communication system. The biggest limitation with *ClassDojo* was there are character limits (255 characters) for messages. When communicating important information with parents, one needs to provide large amounts of details to ensure everyone understood the request or information. *ClassDojo* is not supported by older devices. This is probably not the biggest issue; however, this can cause problems regarding support. *ClassDojo* provides all the same functionality that we had planned to deliver, with the exception of a calendar and platform support..

3.3 *Bloomz*

Bloomz provides all of the same communication features that our PTP would deliver. This application is very similar to *ClassDojo* in the way it would be used as an integral part of the classroom. From the use in the classroom, one can provide

feedback on how your student is doing as well as activities they are working on in school. A few quick searches and this app have received high marks on sites that are comparing similar communication tools. Like ClassDojo, *Bloomz* seems to be a more all-in-one app. These means that these apps would be much more appealing to teachers. A limitation on the *Bloomz* site is that they have a beta version of their application for Windows phones. *Bloomz* support suggests using a browser for the best experience when using a Windows device. There is very little information about using the browser version, so we are unsure if you retain all the features of the app. This would also be the option when using a PC. The PTP that we have designed can be used through the browser. This eliminates these types of platform-dependent apps like *Bloomz*. This will also cut down on the management overhead.

3.4 *ClassTag*

ClassTag provides all of the same communication features that our PTP would deliver with the exception of a class calendar. A class calendar can be generated, but it will need to be synced to other tools like Google accounts, Outlook, etc. Unlike the other three tools, *ClassTag* would provide the platform independence that our app would also provide. *ClassTag* is free and utilizes both browser and apps to connect parents. This will create more overhead hours devoted to maintenance and upkeep. However, ads are displayed in exchange for reward points. We believe this system is only displayed for teachers, but we could not find any information if the ads are used on parents. Their site says that this system helps to offset the costs of maintaining the site and helps keep *ClassTag* free. *ClassTag* is more closely related to our proposed PTP than the other three tools. *ClassDojo* and *Bloomz* attempt to be more integrated within the classroom. We feel that those two tools will run into legal troubles and lose customers because of controversy revolving around collecting data on children and the over focus of technology in the classroom.

ClassDojo and *Bloomz* are also more mature in their development and aim to be an all-in-one tool. This is very appealing, especially for teachers. However, we believe they will be subject to more risk as they are competing as for-profit companies in a non-profit environment. Also, since they are more greatly integrated in the classroom, we also feel that there will be a debate whether these tools meet curriculum requirements. Both areas run the risk of losing support of school systems in our opinion.

4 System Implementation

The goal for Parent-Teacher Portal (PTP) is to provide a forum for teachers and parents to collaborate and be able to provide the highest possible level of learning for students. The challenge is that the information that is delivered by teachers will be

received by parents, not students. To ensure that our idea was grounded in evidence, a series of interviews with actual teachers of elementary age students was also conducted. While performing risk analysis for the PTP, we have identified pertinent risks for our proposed system: system breach and privacy/loss of information.

We have secured the domain name <http://www.parentteacherportal.com> for our portal from Google domains. We have developed webpages using HTML, CSS, and JavaScript codes. For the webpage icons, we are using Bootstrap Glyphicons. Development tools Brackets and Eclipse IDE are used, and for database MySQL Community Edition is used, integrated with webpage using Java servlets pages.

The PTP being hosted and accessed through the Internet will be exposed to all the security vulnerabilities that are being associated with similar sites and technologies. Since we use HTML, CSS, and SQL and each of these tools have past, present, and future vulnerabilities, so to mitigate these vulnerabilities, we used the most up-to-date versions of these software tools and continue to install known good patches. Additionally, we hosted our portal through GitHub for initial deployment. GitHub has put in place several safeguards to protect data in transit during login, and they perform internal and external audits. Privacy and the protection of personal identifying information are a very important area of contention. The data that will be used by our application will consist of names and phone numbers stored in a database and will be transmitted. To mitigate this risk, we limit the types of data that will be needed to create a parent account and the identifying information of their children. This data will be limited to their phone number and first and last name. Secure protocols will be used when data is sent to and from the PTP, and data will remain private using secure coding standards in Java and HTML. The root password for MySQL shall be changed and will not be used for system functions such as running queries. In addition, the MySQL Enterprise version provides the ability to encrypt data at rest encryption in the database using Transparent Data Encryption (TDE). If required, we can upgrade from the free MySQL to the Enterprise version.

The risk of our target audience not using or liking our PTP is one of our biggest risks. The concern regarding parent involvement was mentioned in our stakeholder interviews and comes from the belief that parents that are engaged will be early adopters of the portal, but uninvolved parents will continue to not participate. To mitigate this risk, we will need to develop a deployment plan. This plan shall facilitate and encourage parent involvement. The PTP personnel shall work with the school to host an open house to familiarize parents and introduce the benefits of the portal. During this time, local administrators can assist parents with the processes for setting up and accessing the application functionality on their devices. During the implementation phase and during the beginning of the school year, local administrators will be available during times when parents are more likely to be present on school grounds, for example, when dropping off or picking up their children.

5 Functional and Non-functional Requirements

Parent-Teacher Portal (PTP) app must work on all web and tablet devices. User interface must be consistent on all devices. In addition, the app must also meet the following functional requirements.

1. *Class group creation:* The teacher must first create a class group with name of her choice. The class group will be assigned a system-generated unique ID on the application.
2. *Adding parent\guardian to the class group:* Teacher must be able to register parent\guardian for the application through a valid phone number. There will be an option to register at maximum two contacts per student. After the teacher has added the parents to the group, a short introduction email and text will be sent to the phone numbers. The introduction will contain a link to the web portal and instructions how to register on the portal.
3. *Parents registration:* Parents will receive email from the class teacher with link to create account. The student's ID number will be the verification code to register into the app.
4. *Send Message:* Parents and teacher shall be able to send instant message to each other. They will be notified when a message is successfully delivered to the recipient by displaying a tick sign next to the message sent.
5. *Send Attachments:* Teacher shall be able to send attachments with messages.
6. *Group Email:* Teacher shall be able to send messages to the class group.

6 Usability

PTP usability is the key factor ensuring early and widespread acceptance by both parents and teachers. For this reason, the most heavily used portions of this application are tested rigorously to ensure continuity throughout the application. Security is ensured by following best practices to ensure the privacy and security of all user information during transit and at rest. Code is written securely, data is encrypted during transit, and repeated tests to ensure security measures are in place. Figure 1 shows the link between various functionalities as data flow diagram.

7 User Interface

1. *Home Screen:* The three options available on this screen are Login, Customer Support, and FAQ. The Login button directs the stakeholders to the Login screen and function.
2. *Login Screen/Function:* For the demo, the login and registration are left open. The registration and login functions of the portal are operational.

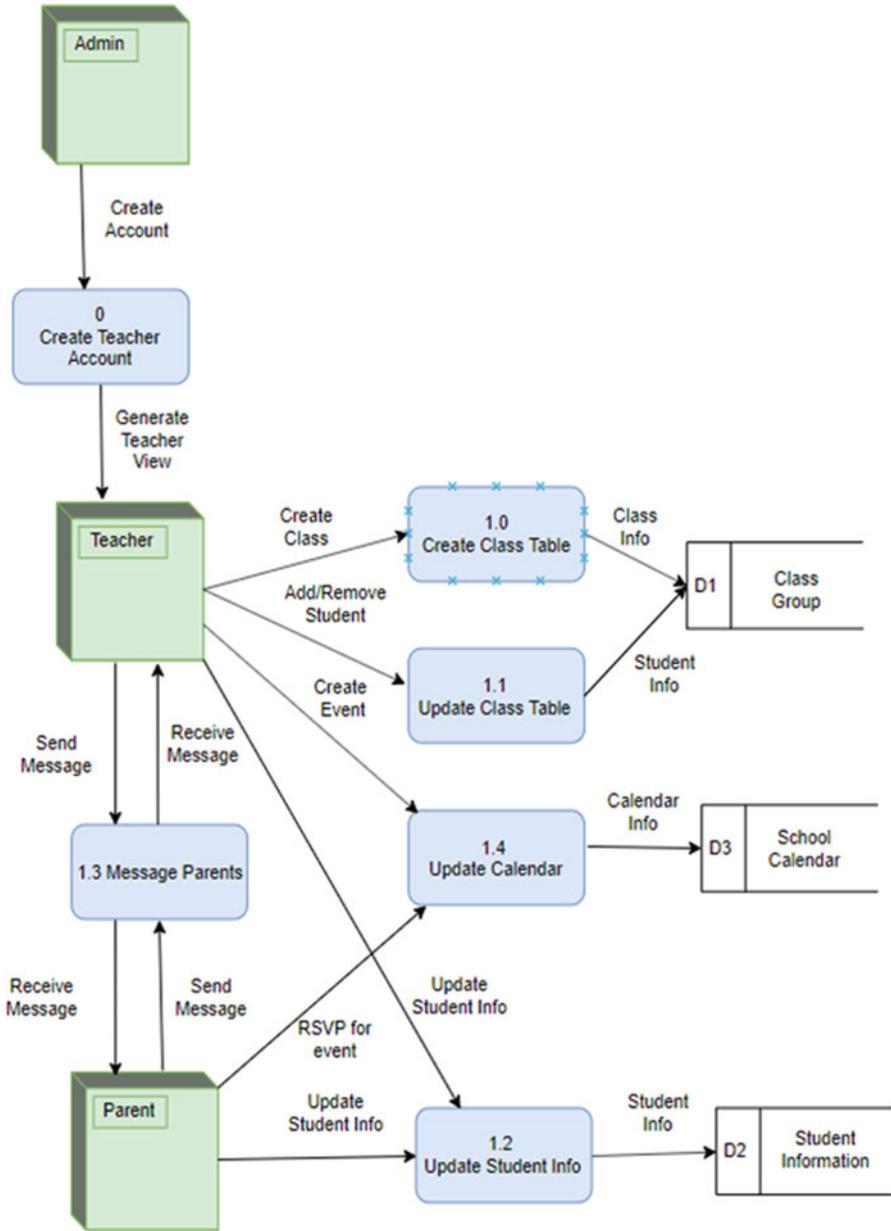


Fig. 1 Data flow diagram

3. *Teacher Home Screen*: The backdrop of the teacher homepage is flexible so that teachers can frequently change these real-world boards to reflect seasons and holidays.

4. *Class Group Screen/Function*: The title banner changed from “Class Groups Assigned” to “Classes Assigned.” It provides the ability to view student details and to see parent contact information.
5. *Email Screen/Function*: The emails shown in this area will be separated by the various classes a teacher is teaching and a collection of all parent emails. It is a filter option that would separate the emails into class groups.

8 Conclusion and Future Recommendation

The main goal of the PTP application was to streamline communication between elementary school teachers and the parents of their students. When interviewing several teachers, they stated that the hardest part of their job can be communicating with parents directly about how their child is doing. The PTP application allows teachers to create class groups with all the contact information of their student’s parents in one location. With that information, they can send a mass class email about a reminder or directly communicate with a parent about a child’s issues in school. Parents can also see the information of other students in their child’s class allowing them to coordinate and work with other parents for school events or volunteering in class. The use of the PTP application should significantly help make communication easier between elementary school teachers and the parents of their students.

We have completed most of the functionality that we set out to create at the beginning. The biggest part that we pulled back on was including a translation service for our application. Along with communication being the hardest part for elementary school teachers, communicating with English language learners (ELLs) can be a barrier. This would be the first and highest priority for future release of our application. In addition, future functionalities are translation service, calendar, send text, class group collaboration for parents, chat feature, discussion board for parents, and graphic design customization.

References

1. J.D. Finn, *School Engagement & Students at Risk* (National Center For Education Statistics, Buffalo) Retrieved on 12 May 2019. Retrieved from <https://nces.ed.gov/pubsearch/pubsinfo.asp?pubid=93470>
2. W. Jeynes, *Parental Involvement and Academic Success* (London, 2010) Retrieved on March 2020. <https://ebookcentral.proquest.com/lib/nu/reader.action?docID=574576>
3. L. Flanagan, *What Can Be Done to Improve Parent-Teacher Communication?* (KQED, 2015)
4. B. Mariconda, *Easy and Effective Ways to Communicate with Parents: Practical Techniques and Tips for Parent Conferences, Open Houses, Notes Home, and More that Work for Every Situation* (Scholastic, 2003)

5. L. Patsalides, *Effective Parent Teacher Communication Ideas for Success: Advice for New Teachers* (Bright Hub Education, 2019)
6. R. Keyes, Parent-teacher partnerships: A theoretical approach for teachers, in *Issues in Early Childhood Education: Curriculum, Teacher Education, & Dissemination of Information*, (Proceedings of the Lilian Katz Symposium, Champaign, 2000)
7. J. Eulberg, 4 Communication Tools to Energize the Parent-Teacher Relationship. Retrieved from on Mar 2020 <https://www.wgu.edu/heyteach/article/4-communication-tools-energize-parent-teacher-relationship1808.html>. Western Governors University

Part IV
Foundations of Computer Science:
Architectures, Algorithms, and
Frameworks

Exact Floating Point



Alan A. Jorgensen and Andrew C. Masters

1 Introduction: IEEE Standard Floating Point

Floating point was used for representing and operating on real numbers in computers starting with the Zuse Z4 computer in 1942. But there was no standard. At the instigation of Professor Emeritus William Morton Kahan, the first standard for floating point, IEEE 754, was published in 1985 (now identified as IEEE 754-1985) by the Institute of Electrical and Electronics Engineers (IEEE). The current version of the floating-point standard is ISO/IEC/IEEE 60559 [1].

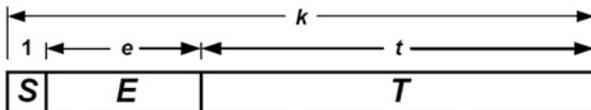
To represent real numbers, standard floating point uses a data structure based on scientific notation, as shown in Fig. 1. This standard floating-point format includes representations for the sign (S), the exponent (E), and the fraction (T).

The sign S is a single bit representing the sign of the value represented, the exponent E is the offset exponent of length e , and the fraction T is the significand of length t . The length k is the overall length of the representation. The real number encoded by this formulaic representation is, in most cases, an approximation, which introduces error. Amplification of this error causes concern about the accuracy of the final result.

The IEEE floating-point standard defines “precision” as “the maximum number, p^{SFP} , of significant digits that can be represented in a format, or the number of digits to that [sic] a result is rounded” [1]. Using the IEEE standard floating-point definition of p^{SFP} , in binary format $p = t + 1$ because of the hidden bit. Instead of merely identifying the number of significant digits that can be represented, bounded floating point provides an enhanced p^{BFP} that represents the actual number of digits (bits) that are significant (have meaning), and the new variable D will represent

A. A. Jorgensen (✉) · A. C. Masters
True North Floating Point, Las Vegas, NV, USA
e-mail: aj@truenorthfloatingpoint.com; andrew@truenorthfloatingpoint.com

Fig. 1 Standard floating-point format number of bits in the significand



the number of digits (bits) of the representation that are NOT significant, where $D = t + 1 - p^{\text{SFP}}$. In IEEE standard representation, the value of D is not known, nor is the precision, p^{BFP} , the actual number of significant bits.

The value of the standard representation is shown in (1).

$$-1^S \cdot (1 + T/2^t) \cdot 2^{E-O} \text{ or } -1^S \cdot ((T + 2^t) / 2^t) \cdot 2^{E-O} \tag{1}$$

where S , T , t , and E are defined above and O is the exponent offset. Offset O is nominally $2^{e-1}-1$.

Most real values (the results from floating-point operations) cannot be represented exactly in standard floating point [2] nor in any fixed number of digits. In bounded floating point, the value is defined to be represented “exactly” when the error between the real value and the floating-point representation is less than $\frac{1}{2}$ units in the last place (ulps) as implied in [2]. In other words, for a given p^{SFP} , there is no other floating-point representation which is closer to the real value.

However, IEEE standard floating point has no mechanism for indicating that the representation of a value is exact. Thus, when only using standard floating point, there is no intrinsic means at present of determining the accuracy of a standard floating-point result. But knowing that a computation is sufficiently correct is important and sometimes vital, particularly in large complex critical computations like weather forecast modeling and other predictive modeling.

Bounded floating point provides that knowledge. Bounded floating point answers the questions that standard floating point alone cannot, such as:

- Is the result “exact”?
- How many significant digits are there in an inexact result?
- Is the result sufficiently accurate?
- Is the result precisely zero?

2 Bounded Floating Point

Recently issued US patents on bounded floating point define a mechanism that calculates and saves the range of error associated with a standard floating-point value [3, 4]. As shown in Fig. 2, bounded floating point extends the standard floating-point representation by adding an error information field identified as the “bound” field, B . Various sizes of formats are selected by the determination of the various field widths where “ k ” is the total format size or floating-point word length. For example, in 80-bit bounded floating point, $k = 80$.

Fig. 2 Bounded floating-point format

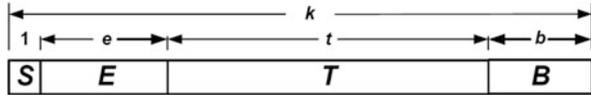
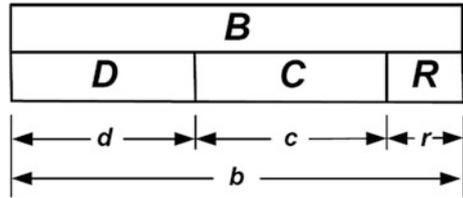


Fig. 3 Format of the bound field B



The bounded floating-point system, implemented in hardware, software, or a combination of the two, calculates and saves the range of error associated with a standard floating-point value, thus retaining and calculating the number of significant digits. It does this by using the bound field *B*.

The bound field *B* contains subfields (*D*, *C*, *R*, see Fig. 3) to retain error information provided by prior operations on the represented value, but the field of primary importance is the “lost bits” field, *D*. This field identifies the number of bits in the represented value that are no longer significant. If the value is exact, the value of the *D* field will be zero.

With the exception of zero detection, bounded floating point retains the exception features of standard floating point. Bounded floating point exactly identifies zero when the significant remaining bits are all zero.

Bounded floating point provides a means of identifying the required number of significant bits (or decimal digits) required in a bounded floating-point calculation. (A default value may be used for the required number of significant bits, or the programmer may specify the required number.) When a result lacks this required number of significant bits, bounded floating point identifies it. A result that lacks the required number of significant bits is represented with the “quiet” not-a-number representation, “qNaN.sig,” indicating excessive loss of significance.

Under program control, bounded floating point will provide a “signaling” not-a-number representation, “sNaN.sig,” when a specified bounded floating-point value does not meet a specified precision requirement. Upon initiation of this command, a specific result is tested to verify that it has the required number of significant digits.

The interval defined by bounded floating point is given by (2) as follows:

$$-1^S \bullet \left((T + 2^t) / 2^t \right) \bullet 2^{E-O} \dots - 1^S \bullet \left((T + 2^t + 2^{D-1}) / 2^t \right) \bullet 2^{E-O} \quad (2)$$

This is the same as standard floating point except that the term 2^{D-1} provides the upper bound, where *D* is the logarithm of the number of bits that are no longer significant. Importantly, when *D* is zero, this indicates that there are no insignificant digits; the bound is 1/2 ulp (2^{D-1} when *D* = 0 is 1/2) and the value of the error is less than or equal to the bound, which is the definition of an “exact” representation.

In Fig. 3, the R field of the bound is the summation of the most significant bits lost during the final truncation and is functionally equivalent to the guard and rounding bits of the standard floating-point operations. The equivalent of the “sticky bit” occurs when the remainder of the truncated bits is not zero so that one is added to the value of R , the rounding error field of the bound field. Addition to the R field carries into the C field, which is the rounding error count in units-in-the-last-place (ulps).

Range information includes the number of bits in the representation that are no longer of value (insignificant) and, therefore, are referred to as the “lost bits” (D). The number of lost bits is the logarithm of the upper bound (furthest from zero) of the error in the value represented. The lost bits include the accumulated contributions from both cancellation and rounding errors.

When the based two logarithm of the resulting sum of the rounding error, C , is greater than or equal to the resulting lost bits, D , the lost bits, D , is increased by one, and the rounding error sum, C , is set to zero. Carries out of the C field are added to the D field.

This calculation provides a worst-case interval in which the real value represented exists. The actual precision, p^{BFP} , of the value represented is no greater than $t + 1 - D$. When the value of the D field for a represented value is zero, then the representation is “exact” as defined above.

The truncated floating-point value (round to zero) is the lower bound, and the upper bound is determined by the addition of the error determined by the lost bits, D (the real value represented), $V \in \mathbb{R}$, is absolutely contained in the interval of (2).

The midpoint is determined by (3), as follows:

$$-1^S \bullet \left(\left((T + 2^t + 2^{D-2}) / 2t \right) \bullet 2^{E-O} \right) \quad (3)$$

Bounded floating point allows accuracy of the source of real values, measured or entered, to be specified. External data sources provide data with intrinsic error; for example, keyboard data entry with a limited precision input field or an industrial sensor that provides fewer significant bits than that required by the precision of the floating-point format in use.

According to Ashenhurst and Metropolis [5]:

It is convenient, and by now more or less traditional, to distinguish three sources of error, designated generated, inherent and analytic. Generated error reflects inaccuracies due to the necessity of rounding or otherwise truncating the numeric results of arithmetic operations, inherent error reflects inaccuracies in initially given arguments and parameters, and analytic error reflects inaccuracies due to the use of a computing procedure which calculates only an approximation to the theoretical result desired.

Bounded floating point permits the representation of inherent error (inaccuracies in input parameters). If the number of significant digits in the value provided is limited, then bounded floating point can accurately represent that number by subtracting the number of bits required to represent that number from p^{SFP} to obtain D (the number of lost bits), which are then carried throughout the calculation.

Bounded floating point can manage generated error by calculating the number of bits that are lost due to “the necessity of rounding or otherwise truncating the numeric results of arithmetic operations.”

Additionally, bounded floating point can be used in conjunction with implementations of the current floating-point standard. Conversion between the two formats can be accomplished when needed, which allows continued use of existing software that is dependent upon the current floating-point standard. However, error information will be lost when converting from bounded floating point to standard floating point.

3 Similar Floating-Point Numbers

Catastrophic cancellation occurs when subtracting similar numbers when error already exists [2, 6, p. 124, 7, 8, pp., 10–11, 9].

“Similar numbers” can be defined by (4) that describes the loss of significant digits, as suggested by [10].

$$\begin{aligned}
 D &= \text{Log}_2(z); \text{ iff } V \bullet z / (z + 1) > M/S \\
 &\geq V \bullet (z - 1) / z \text{ and } z \subset \left\{ 2^i, i = 3..p^{\text{SFP}} \right\}
 \end{aligned}
 \tag{4}$$

where D is the resulting number of insignificant (lost) bits, M is the minuend, S is the subtrahend, V is the represented floating-point value ($V \in \mathbb{R}$), p^{SFP} is the number of bits in the significand including the hidden bit, and M or S is inexact. Note that for n less than 3, when guard digits are applied, the result will be “exact” [8, pp. 48–50, 11, p. J23].

The error, as represented by bounded floating point, due to the cancellation is no greater than 2^{D-1} .

Bounded floating point uses the value of D to determination the number of significant digits of a value. This is done by taking the value of p^{SFP} , which identifies the highest number of significant digits that can be represented in a format, and subtracting D , which identifies the number of insignificant or lost bits. The result, which is the enhanced p^{BFP} , of the subtraction establishes the number of significant digits in an “exact” or inexact result.

Also, by using D a determination can be made as to whether the result is sufficiently accurate. The required number of significant digits is known (either by use of a default value or by programming a number required). If the result p^{BFP} (the number of actual significant digits) is less than the required number of significant digits, the number is inexact. If the resulting p^{BFP} is equal to or greater than the required number of significant digits, the number is sufficiently accurate.

Another advantage of having the value of D known and available for use is that a determination can be made as to whether a result is precisely zero. Knowing the number of significant digits in a number allows bounded floating point to compare

the number of digits that are significant against the number of leading zeros in the result. If the significant digits of the result are all zero, the result is determined to be significantly zero. This is in contrast to standard floating point, in which all digits of the result must be zero.

Consequently, bounded floating point enables a determination of the “exactness” of a value, discloses the actual number of significant bits, and ascertains when a result is precisely zero, none of which are available without using bounded floating point.

4 Exact and Inexact Subtraction

Subtraction is “exact” when the subtrahend and minuend have no rounding error, as stated by Goldberg in “What Every Computer Scientist Should Know About Floating-Point Arithmetic” [11]. However, when inexact, but similar, values are subtracted, rounding error will cause catastrophic cancellation with a corresponding loss of significant digits , [8 , p. 11, 11].

Table 1 shows subtraction of similar values and demonstrates catastrophic cancellation, which occurs when the two values (minuend A and subtrahend B) to be subtracted are similar as defined in (4).

For a test case we have selected $A - B$ where $A = 10,000,000,000 \cdot \pi$ (scaled for ease of representation of the result), selected $z = 4,294,967,296 (2^{32})$, and used $B = A \cdot (z-1)/z$ from (4). The selection of 2^{32} indicates that there are 32 lost bits in this example.

To assure that no error was introduced by standard floating point, Table 1 provides these values as computed by Mark Mason’s High Precision Calculator that was set for “High Precision” [12, 13], which provides a surplus number of digits (not the limited number of digits of the 64-bit or 128-bit floating point) for the calculations. Consequently, the values shown are “exact” values.

Using the example in Table 1, the value of B , which is **31415926528.583341988 . . .** is subtracted from the value of A , which is **31415926535.897932384 . . .** We know that standard floating-point calculations are constrained to a limited number of digits. If we consider this calculation as limited to nine decimal digits, when the

Table 1 Subtraction of similar values

High precision results
$A = 10^{10} \cdot \pi = 31415926535.89793238462643383279502884197169399375$
$z = 2^{32} = 4,294,967,296,496,729$
$(z - 1)/z = 0.999999997671693563461303710937$
$B = A \cdot (z - 1)/z = 314159265$ 28.5833419882906354275383398204227325084871797295159194618463516235351562
$A - B =$ 7.3145903963357984052566890215489614852638202704840805381536483764648437

subtraction is performed, the first nine digits will all be zero (will cancel out), leaving no significant digits – clearly showing catastrophic cancellation.

This potential for the lack of significant digits in standard floating point is not new. It has been known from at least 1952 when an early floating-point patent [14] by IBM explicitly stated “. . . under some conditions, the major portion of the significant data digits may lie beyond the capacity of the registers. Therefore, the result obtained may have little meaning if not totally erroneous.” Bounded floating point can be used to clearly identify when there is a lack of significant digits.

Table 2 presents the comparison of “exact” and inexact values differing by one binary ulp calculated with 64-bit standard floating point and 128-bit standard floating point.

The first row shows that the decimal representation of the value of *A*, using 64-bit floating point, is 31415926535.897930, while the second row shows that after adding only one ulp of error to *A* the decimal equivalent of *A* + 1 is 31415926535.897934.

The second and third rows show the decimal equivalents of *A* and *A* + 1 using 128-bit floating point.

Table 2 shows the effect of even a very small error of only one ulp, which creates an inexact value. When similar values are subtracted, cancellation [2] occurs, and the one-bit error is multiplied exponentially. This is a standard floating point hidden and unknown error, but this error is revealed in bounded floating point.

Table 3 demonstrates “exact” and inexact calculations in 64-bit, 128-bit, and 80-bit bounded floating-point calculations. This table illustrates the results of adding a one ulp error injected into the “*A*” values by adding one to the significand field (*T*) of the 64-bit and 128-bit standard floating-point values and adding one to the lost bits field (*D*) of the bounded floating-point value. Overflow is avoided by the selection of test values.

Table 2 High precision similar values calculation

Represented value	Decimal representation of value
64-bit FP <i>A</i>	31415926535.897930
64-bit FP <i>A</i> + 1 ulp	31415926535.897934
128-bit FP <i>A</i>	31415926535.8979323846264338327950 28075364135510
128-bit FP <i>A</i> + 1 ulp	31415926535.8979323846264338327950 31384086585722

Table 3 Exact and error-injected values – 64-bit results

Exact 64-bit result	7.31459045410156250
Inexact 64-bit result	7.31459 426879882810
Exact 128-bit result	7.314590396335798405256687972038204802
Inexact 128-bit result	7.3145903963357984052566 91280760655014
Exact BFP 80-bit result	7.314590454101562
Inexact BFP 80-bit result	7.314590

Tests were performed using IEEE standard 64-bit floating point, 128-bit standard floating point, and 80-bit bounded floating point (BFP), as seen in Table 3. The 80-bit bounded floating-point model consists of S , E , and T , which are identical to 64-bit standard floating point with a 16-bit bound field, where $d = 6$, $c = 6$, and $r = 4$. These values are chosen so that the total width, b , is a multiple of 8 bits and d (the length of the lost bits field) satisfies $d > \log_2(t + 1)$ to ensure that a loss of all significant bits can be represented. The value for c (the accumulated rounding error in ulps) is chosen such that exponential growth rate of the loss of significant bits due to rounding error will not exceed 2^n where $n = 1/2^c$, or, in this case, $1/64$. The width r of the rounding error field R is chosen to round the width b of the bound field, B , up to the nearest multiple of 8-bits.

The bold and underlined digits of the 64-bit inexact results are those digits that differ from the same digit positions of the “exact” 64-bit calculation. Similarly, the bold and underlined digits of the 128-bit inexact result differ from the “exact” 128-bit calculation. Table 3 makes it easy to see the difference in “exact” and inexact values. And when similar numbers with inexact values (such as may arise from error in earlier calculations or error from input with limited significant digits) are subtracted using floating point, these errors can multiply exponentially due to catastrophic cancellation.

5 Conclusions

Floating-point cancellation errors that occur during subtract operations on inexact operands are detectable and measurable under bounded floating point though they are invisible in IEEE standard floating-point results.

Bounded floating point answers the following questions that standard floating point cannot:

- Is the result “exact”?
 - An “exact” floating-point result, defined as a result that has error within $+ or - \frac{1}{2}$ units in the last place (ulps), is shown by using the value of D .
- How many significant digits are there in an inexact result?
 - The number of digits known to be insignificant, D , is subtracted from the possible number of significant digits, which is known from p^{SFP} .
- Is the result sufficiently accurate?
 - The number of significant digits needed is merely compared to the number of significant digits that is known by use of bounded floating point.
- Is the result precisely zero?
 - When bounded floating point determines the number of significant digits of the result is all zero, then the result is significantly zero.

Thus, bounded floating point precisely defines whether the real value represented is “exact” for all digits provided and, if not, defines the number of significant digits. And bounded floating point provides notification if the result is not significantly accurate.

The bounded floating-point methodology provides greater assurance that complex mission critical computations provide results sufficient to successfully complete that mission. Therefore, it is recommended that bounded floating point should be required for all mission critical systems to avoid catastrophic failures due to accumulated floating-point error.

References

1. ISO/IEC/IEEE 60559, *Information Technology – Microprocessor Systems – Floating-Point Arithmetic* (Institute of Electrical and Electronics Engineers, Piscataway, 2011)
2. D. Goldberg, What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.* **23**(1), 5–48 (1991)
3. A. A. Jorgensen, Apparatus for calculating and retaining a bound on error during floating point operations and methods thereof. US Patent No. 9,817,662, 14 Nov 2017
4. A. A. Jorgensen, Apparatus for calculating and retaining a bound on error during floating point operations and methods thereof. US Patent No. 10,540,143, 21 Jan 2020
5. R.L. Ashenurst, N. Metropolis, Error estimation in computer calculation. *Am Math Monthly*, Part 2: *Comp Comp* **72**(2), 47–58 (1965)
6. J.-M. Muller, F. de Dinechin, C.P. Jeannerod, V. Lefevre, G. Melquiond, N. Revo, D. Stehle, S. Torres, *Handbook of Floating-Point Arithmetic* (Birkhauser, Boston, 2010)
7. N.J. Higham, *Accuracy and Stability of Numerical Algorithms* (SIAM, Philadelphia, 1996), p. vii–xxviii, 1–688
8. W. M. Kahan, A Logarithm Too Clever by Half, 2004. [Online]. Available: <http://people.eecs.berkeley.edu/~wkahan/LOG10HAF.TXT>. Accessed 26 Feb 2019
9. W.E. Ferguson Jr., Exact computation of a sum or difference with applications to argument reduction, in *Proceedings of the 12th IEEE Symposium on Computer Arithmetic*, (Bath, 1995)
10. W.M. Kahan, Desperately needed remedies for the undebuggability of large floating-point computations in science and engineering, in *IFIP/SIAM/NIST Working Conference on Uncertainty Quantification in Scientific Computing*, (Boulder, 2011)
11. D. Goldberg, What every computer scientist should know about floating-point arithmetic. *ACM Comput. Surv.* **23**(1), 5–47 (1991)
12. A.A. Jorgensen, A. Masters, R. Guha, Assurance of accuracy in floating-point calculations – A software model study, in *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, (Las Vegas, 2019)
13. N. M. Mason, High Precision Calculator – Freeware, 2017. [Online]. Available: <http://www.markmason.net/hpc/index.htm>. Accessed 28 Feb 2020
14. H. M. Sierra, Floating decimal point arithmetic control means for calculator, United States Patent 3,037,701, 5 June 1962

Random Self-modifiable Computation



Michael Stephen Fiske

1 Introduction

What is computation? This question usually assumes that the Turing machine¹ [23] is the standard model [18, 19, 21]. We reexamine this question with a new model, called the *ex-machine* [12]. This model adds two special instructions to the Turing machine instructions. The name *ex-machine* comes from the Latin term *extra machinam* because the *ex-machine* computation is a non-autonomous dynamical system [10] that may no longer be considered a machine.² The *meta instruction* adds new states and new instructions or can replace instructions. The *random instruction* can be physically realized with a quantum random number generator [14, 15]. When an *ex-machine* uses meta and random instructions, its program complexity (machine size [3]) can increase, unlike a lever, pulley, or Turing machine. Two identical *ex-machines* can evolve to different *ex-machines* even when both start executing with the same tape input and initial state.

We combine self-modification and randomness and construct an *ex-machine* $Z(x)$ whose program complexity $|Q||A|$ increases as it executes.³ $Z(x)$'s non-autonomous behavior circumvents the contradiction in an information-theoretic proof [4, 5] of Turing's halting problem. The proof's contradiction depends upon an

¹The conception of the Turing machine was motivated by Hilbert's goal to find a general method for constructing proofs of mathematical theorems [15].

²Each Turing machine is a discrete autonomous dynamical system in \mathbb{C} . See the Appendix.

³ Q and A represent the *ex-machine* states and alphabet, respectively.

M. S. Fiske (✉)
Aemea Institute, San Francisco, CA, USA
e-mail: mf@aemea.org

information-theoretic property: each Turing machine is representable with a finite number of bits that stays constant during the entire execution. Some ex-machines violate this property. $Z(x)$'s circumvention occurs because its meta instructions increase the number of states and instructions in $Z(x)$, based on random information obtained from its random instructions. Hence, the minimal number of bits that represent an ex-machine's evolved program can increase without bound as execution proceeds.

1.1 Related Work—Computation

In [24], the notion of providing an oracle was introduced. Turing stated that *an oracle cannot be a machine* but did not provide a physical basis for its existence. For a summary of various physical realizations that use quantum events to generate random binary outcomes, see [14]. In [7], the following question was asked: *Is there anything that can be done by a machine with a random element but not by a deterministic machine?* They showed for a Turing computable probability p (e.g., $p = \frac{1}{2}$) that any set of output symbols that can be enumerated with positive probability by their probabilistic machine can also be enumerated by a Turing machine. Overall, they were unable to produce Turing incomputable computation when p is Turing computable. In [13], a framework is developed for self-modifying programs, but it does not include randomness and does not address computability. In [11], a parallel machine self-modifies with meta commands and takes quantum random measurements to execute a Turing incomputable black box. Prior hypercomputation models [8, 16, 17] are not physically realizable.

2 The Ex-machine

\mathbb{Z} , \mathbb{N} , and \mathbb{N}^+ are the integers, non-negative integers, and positive integers, respectively. The finite set $Q = \{0, 1, 2, \dots, n-1\} \subset \mathbb{N}$ represents the ex-machine states. As a subset of \mathbb{N} , Q helps specify how new states are added to Q when a meta instruction executes. Let $V = \{a_1, \dots, a_n\}$. The set $A = \{0, 1, \#\} \cup V$ consists of alphabet (tape) symbols, where $\#$ is the blank symbol and $\{0, 1, \#\} \cap V = \emptyset$. In some ex-machines, $A = \{0, 1, \#, Y, N, a\}$, where $V = \{Y, N, a\}$. Sometimes, $A = \{0, 1, \#\}$. Alphabet symbols are scanned from and written on the tape. The tape is a function $T : \mathbb{Z} \rightarrow A$. We say T is *finite* [21], whenever a finite number of tape squares $T(k)$ contain non-blank symbols.

2.1 Standard Instructions

Definition 1 (Execution of Standard Instructions) Standard instructions S satisfy $S \subset Q \times A \times Q \times A \times \{-1, 0, 1\}$ and a uniqueness condition: If $(q_1, \alpha_1, r_1, a_1, y_1) \in S$ and $(q_2, \alpha_2, r_2, a_2, y_2) \in S$ and $(q_1, \alpha_1, r_1, a_1, y_1) \neq (q_2, \alpha_2, r_2, a_2, y_2)$, then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$. Instruction $I = (q, a, r, \alpha, y)$ follows [19]. When the ex-machine is in state q and the tape head is scanning $a = T(k)$ at tape square k , I executes as follows. The ex-machine state moves from state q to state r . Alphabet symbol a is replaced with α so that $T(k) = \alpha$. If $y = -1$ or 1 , the tape head moves one square to the left or right, respectively. If $y = 0$, the tape head does not move.

A Turing machine [23] has a finite set of machine states, a finite alphabet, a finite tape, and a finite set of standard instructions that execute according to Definition 1. An ex-machine that uses only standard instructions is called a *standard machine* and is computationally equivalent to a Turing machine. The Turing machine is the standard mathematical model of computation, realized by digital computers [1].

2.2 Random Instructions

This subsection defines two random axioms and the random instructions. Repeated independent trials are called *quantum random Bernoulli trials* [9] if all trials have a quantum random measurement [14] with only two outcomes and the probability of each outcome stays constant. *Unbiased* means that the probability of both outcomes is the same.

Random Axiom 1 (Unbiased Trials) Quantum random outcome x_i measures 0 or 1. Probability $P(x_i = 1) = P(x_i = 0) = \frac{1}{2}$.

Random Axiom 2 (Stochastic Independence) Prior measurements x_1, \dots, x_{i-1} have no effect on the next measurement x_i . For each $b_i \in \{0, 1\}$, the conditional probabilities satisfy $P(x_i = 1 | x_1 = b_1, \dots, x_{i-1} = b_{i-1}) = \frac{1}{2}$ and $P(x_i = 0 | x_1 = b_1, \dots, x_{i-1} = b_{i-1}) = \frac{1}{2}$.

Definition 2 (Execution of Random Instructions) Random instructions \mathfrak{R} are a subset of $Q \times A \times Q \times \{-1, 0, 1\}$. \mathfrak{R} satisfies uniqueness condition: If $(q_1, \alpha_1, r_1, y_1) \in \mathfrak{R}$ and $(q_2, \alpha_2, r_2, y_2) \in \mathfrak{R}$ and $(q_1, \alpha_1, r_1, y_1) \neq (q_2, \alpha_2, r_2, y_2)$, then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$. When scanning symbol a and in state q , instruction (q, a, r, y) executes as follows:

- (1) Measure bit $b \in \{0, 1\}$ from a quantum random source that satisfies both axioms.
- (2) On the tape, an alphabet symbol a is replaced with a random bit b . Note $\{0, 1\} \subset A$.
- (3) The ex-machine state q changes to state r .

- (4) The tape head moves left if $y = -1$, moves right if $y = 1$, and does not move if $y = 0$.

Example 1 lists a random walk ex-machine; it shows how the random instructions execute and how the ex-machine can exhibit non-autonomous dynamical behavior.

Example 1 (Random Walk Ex-machine) Alphabet $A = \{0, 1, \#, E\}$. $Q = \{0,1,2,3,4,5,6,h\}$ with halting state $h = 7$. There are 3 random instructions $(0, \#, 0, 0)$, $(1, \#, 1, 0)$, and $(4, \#, 4, 0)$.

```

(0, #, 0, 0)      (0, 0, 1, 0, -1)   (0, 1, 4, 1, 1)      ; Comments follow a semicolon.
(1, #, 1, 0)      (1, 0, 1, 0, -1)   (1, 1, 2, #, 1)      ; Resume random walk to the left
of tape square 0
(2, 0, 3, #, 1)   (2, #, h, E, 0)      (2, 1, h, E, 0)
(3, #, 0, #, -1)  ; Go back to state 0.  Number of random 0's = Number of
random 1's.
(3, 0, 1, 0, -1) ; Go back to state 1.  Number of random 0's > Number of
random 1's.
(3, 1, h, E, 0)
(4, #, 4, 0)      (4, 1, 4, 1, 1)   (4, 0, 5, #, -1)    ; Resume random walk to the right
of tape square 0
(5, 1, 6, #, -1) (5, #, h, E, 0)      (5, 0, h, E, 0)
(6, #, 0, #, 1)  ; Go back to state 0.  Number of random 0's = Number of
random 1's.
(6, 1, 4, 1, 1) ; Go back to state 4.  Number of random 1's > Number of
random 0's.
(6, 0, h, E, 0)
    
```

A valid initial tape contains only blank symbols. A valid initial state is 0. At step 1, random instruction $(0, \#, 0, 0)$ measures 0, so it executes $(0, \#, 0, 0, 0)$. At step 3, instruction $(1, \#, 1, 0)$ measures 1, so it executes $(1, \#, 1, 1, 0)$. (Per Definition 2, 0_r means 0 was randomly measured, and 1_r means 1 was measured.) In all executions shown, the tape head is reading the symbol to the right of the space. The sequence of tape symbols shows the tape contents after the instruction in the same row has executed.

First Execution of Random Walk Ex-machine. Steps 1–7.

STATE	TAPE	HEAD	INSTRUCTION
0	### 0###	0	$(0, \#, 0, 0_r, 0)$
1	## #0###	-1	$(0, 0, 1, 0, -1)$
1	## 10###	-1	$(1, \#, 1, 1_r, 0)$
2	### 0###	0	$(1, 1, 2, \#, 1)$
3	#### ##	1	$(2, 0, 3, \#, 1)$
0	### #####	0	$(3, \#, 0, \#, -1)$
0	### 0###	0	$(0, \#, 0, 0_r, 0)$

For the second execution, at step 1, a random measurement returns a 1, so $(0, \#, 0, 0)$ executes as $(0, \#, 0, 1, 0)$. Instruction $(4, \#, 4, 0)$ measures 0, so $(4, \#, 4, 0, 0)$ executes.

Second Execution of Random Walk Ex-machine. Steps 1–7.

STATE	TAPE	HEAD	INSTRUCTION
0	### 1###	0	$(0, \#, 0, 1_r, 0)$
4	####1 ###	1	$(0, 1, 4, 1, 1)$
4	####1 0##	1	$(4, \#, 4, 0_r, 0)$
5	### 1###	0	$(4, 0, 5, \#, -1)$
6	## #####	-1	$(5, 1, 6, \#, -1)$
0	### #####	0	$(6, \#, 0, \#, 1)$
0	### 1###	0	$(0, \#, 0, 1_r, 0)$

The first and second executions show that the execution behavior of the same ex-machine with identical initial conditions may be distinct at two different instances. Hence, the ex-machine is a discrete, non-autonomous dynamical system [10].

2.3 Meta Instructions

This subsection defines the meta instruction and the notion of evolving an ex-machine. The execution of a meta instruction can add new states and new instructions or replace instructions. Formally, the meta instructions \mathfrak{M} satisfy $\mathfrak{M} \subset \{(q, a, r, \alpha, y, J) : q \in \mathfrak{Q}$ and $r \in \mathfrak{Q} \cup \{|\mathfrak{Q}|\}$ and $a, \alpha \in A$ and instruction $J \in \mathfrak{S} \cup \mathfrak{R}\}$. Define $\mathfrak{T} = \mathfrak{S} \cup \mathfrak{R} \cup \mathfrak{M}$, as the set of standard, random, and meta instructions. To help describe how a meta instruction modifies \mathfrak{T} , the *unique state, scanning symbol condition* is defined. For any two distinct instructions in \mathfrak{T} , at least one of the first two coordinates must differ. More precisely, all six of the following uniqueness conditions must hold.

1. If $(q_1, \alpha_1, r_1, \beta_1, y_1)$ and $(q_2, \alpha_2, r_2, \beta_2, y_2)$ both are in \mathfrak{S} , then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$.
2. If $(q_1, \alpha_1, r_1, \beta_1, y_1) \in \mathfrak{S}$ and $(q_2, \alpha_2, r_2, \beta_2, y_2) \in \mathfrak{R}$, then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$.
3. If $(q_1, \alpha_1, r_1, y_1)$ and $(q_2, \alpha_2, r_2, y_2)$ both are in \mathfrak{R} , then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$.
4. If $(q_1, \alpha_1, r_1, y_1) \in \mathfrak{R}$ and $(q_2, \alpha_2, r_2, a_2, y_2, J_2) \in \mathfrak{M}$, then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$.
5. If $(q_1, \alpha_1, r_1, \beta_1, y_1) \in \mathfrak{S}$ and $(q_2, \alpha_2, r_2, a_2, y_2, J_2) \in \mathfrak{M}$, then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$.
6. If $(q_1, \alpha_1, r_1, a_1, y_1, J_1) \in \mathfrak{M}$ and $(q_2, \alpha_2, r_2, a_2, y_2, J_2) \in \mathfrak{M}$, then $(q_1, \alpha_1) \neq (q_2, \alpha_2)$.

Given a valid machine specification, conditions 1–6 assure that there is no ambiguity on what instruction to execute. The execution of a meta instruction preserves conditions 1–6.

Definition 3 (Execution of Meta Instructions) Meta instruction (q, a, r, α, y, J) executes as follows:

- (1) The first five coordinates (q, a, r, α, y) are executed as a standard instruction according to Definition 1 with one caveat. State q may be expressed as $|\mathfrak{Q}|-c$ and state r may be expressed as $|\mathfrak{Q}|$ or $|\mathfrak{Q}|-d$, where $0 < c, d \leq |\mathfrak{Q}|$. When (q, a, r, α, y) is executed, if q is expressed as $|\mathfrak{Q}|-c$, the value of q is instantiated to the current value of $|\mathfrak{Q}| - c$. Similarly, if r is expressed as $|\mathfrak{Q}|$ or $|\mathfrak{Q}|-d$, the value of state r is instantiated to the current value of $|\mathfrak{Q}|$ or $|\mathfrak{Q}| - d$, respectively.
- (2) Instruction J modifies \mathfrak{T} , where J has the form $J = (q, a, r, \alpha, y)$ or $J = (q, a, r, y)$. If $\mathfrak{T} \cup \{J\}$ satisfies the unique state, scanning symbol condition, then \mathfrak{T} is updated to $\mathfrak{T} \cup \{J\}$. Otherwise, there is an instruction I in \mathfrak{T} whose first

two coordinates q and a equal instruction J 's first two coordinates. In this case, instruction J replaces instruction I in \mathcal{J} , and \mathcal{J} is updated to $\mathcal{J} \cup \{J\} - \{I\}$.

Remark 1 (Ex-machine Instructions are Sequences of Sets) This remark clarifies the definitions of machine states, standard, random, and meta instructions. The machine states are formally a sequence of sets. When the notation is formally precise, the machine states are expressed as $\Omega(m)$, where m indicates that the m th computational step has executed. The standard, random, and all ex-machine instructions are also sequences of sets, represented as $\mathfrak{S}(m)$, $\mathfrak{R}(m)$, and $\mathfrak{J}(m)$, respectively. Usually, index m is not shown in expressions Ω , \mathfrak{S} , \mathfrak{R} , \mathfrak{M} , or \mathfrak{J} .

Example 2 shows how to add an instruction to \mathcal{J} and how to instantiate new states in Ω .

Example 2 (Adding New States and Instructions) Consider a meta instruction $(q, a_1, |\Omega|-1, \alpha_1, y_1, J)$, where $J = (|\Omega|-1, a_2, |\Omega|, \alpha_2, y_2)$. After instruction $(q, a_1, |\Omega|-1, \alpha_1, y_1)$ executes, this meta instruction adds a new state $|\Omega|$ to the states Ω and adds instruction J , instantiated with the current value of $|\Omega|$. For clarity, states are red and alphabet symbols are blue. Set $\Omega = \{0, 1, 2, 3, 4, 5, 6, 7\}$. Set $A = \{\#, 0, 1\}$. An initial configuration is shown below.

State	Tape
5	##11 01##

Meta instruction $(5, 0, |\Omega| - 1, 1, 0, J)$ executes with values $q = 5, a_1 = 0, \alpha_1 = 1, y_1 = 0, a_2 = 1, \alpha_2 = \#,$ and $y_2 = -1$. Note $J = (|\Omega|-1, 1, |\Omega|, \#, -1)$. Since $|\Omega| = 8$, instruction $(5, 0, 7, 1, 0)$ is executed. Also, standard instruction $J = (7, 1, 8, \#, -1)$ is added as a new instruction. The instantiation of $|\Omega| = 8$ in J adds state 8; the states are updated to $\Omega = \{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. After $(5, 0, |\Omega| - 1, 1, 0, J)$ executes, the new ex-machine configuration is shown below.

State	Tape
7	##11 11##

Now, the ex-machine is scanning a 1 and lying in state 7, so the standard instruction $J = (7, 1, 8, \#, -1)$ executes. (Note that J was just added to the instructions.) After J executes, the new configuration is shown below.

State	Tape
8	##1 1#1##

Remark 2 (Self-reflection of $|\Omega|$) Consider an ex-machine \mathfrak{X} with a meta instruction I containing symbol $|\Omega|$. The instantiation of $|\Omega|$ invokes *self-reflection* about \mathfrak{X} 's current number of states, at the moment when \mathfrak{X} executes I . This type of self-reflection can be physically realized.

Definition 4 (Simple Meta Instructions) $(q, a, |\Omega|-d, b, y), (q, a, |\Omega|, b, y), (|\Omega|-c, a, r, y), (|\Omega|-c, a, |\Omega|-d, b, y),$ or $(|\Omega|-c, a, |\Omega|, b, y)$ are valid expressions for simple meta instructions, where $0 < c, d \leq |\Omega|$. Symbols $|\Omega|-c, |\Omega|-d,$ and $|\Omega|$ instantiate to a state based on the value of $|\Omega|$ when the simple meta instruction executes.

Herein, ex-machines self-reflect only with symbols $|\Omega|-1$ and $|\Omega|$.

Example 3 (Execution of Simple Meta Instructions.) $A = \{0, 1, \#\}$ and $\Omega = \{0\}$.
 Instructions $(|\Omega|-1, \#, |\Omega|-1, 1, 0)$ $(|\Omega|-1, 1, |\Omega|, 0, 1)$

STATE	TAPE	HEAD	INSTRUCTION	NEW INSTRUCTION
0	# 1##	0	(0, #, 0, 1, 0)	(0, #, 0, 1, 0)
1	#0 ##	1	(0, 1, 1, 0, 1)	(0, 1, 1, 0, 1)
1	#0 1#	1	(1, #, 1, 1, 0)	(1, #, 1, 1, 0)
2	#00 #	2	(1, 1, 2, 0, 1)	(1, 1, 2, 0, 1)

With an initial blank tape and starting state of 0, four computational steps are shown above. At step 1, \mathfrak{X} scans # and lies in state 0. Since $|\Omega| = 1$, a simple meta instruction $(|\Omega|-1, \#, |\Omega|-1, 1, 0)$ instantiates to $(0, \#, 0, 1, 0)$ and executes. At step 2, \mathfrak{X} scans 1 and lies in state 0. Since $|\Omega| = 1$, $(|\Omega|-1, 1, |\Omega|, 0, 1)$ instantiates to $(0, 1, 1, 0, 1)$, updates $\Omega = \{0, 1\}$, and executes $(0, 1, 1, 0, 1)$.

Definition 5 (Finite Initial Conditions) Ex-machine \mathfrak{X} has finite initial conditions if the 4 conditions hold before \mathfrak{X} 's instructions are executed: (1) The number of states $|\Omega|$ is finite. (2) The number of alphabet symbols $|A|$ is finite. (3) The number of instructions $|\mathfrak{I}|$ is finite. (4) The tape is finite.

An ex-machine's initial conditions are analogous to a differential equation's boundary value conditions. Remark 3 assures that the ex-machine computation is physically plausible.

Remark 3 (Finite Initial Conditions) If the machine starts its execution with finite initial conditions, then after the machine has executed l instructions for any positive integer l , the current number of states $\Omega(l)$ is finite and the current set of instructions $\mathfrak{I}(l)$ is finite. Also, tape T is still finite, and the number of quantum random measurements obtained is finite.

Proof The execution of one meta instruction adds at most one new instruction and one new state to Ω . Using induction, Remark 3 follows from Definitions 1, 2, 3, and 5.

An ex-machine can evolve from a prior computation. *Evolution* is useful: random and meta instructions can increase an ex-machine's complexity via self-modification.

Definition 6 (Evolving an Ex-machine) Let $T_0, T_1, T_2 \dots T_{i-1}$ each be a finite tape. Consider an ex-machine \mathfrak{X}_0 with finite initial conditions. \mathfrak{X}_0 starts executing with tape T_0 and evolves to ex-machine \mathfrak{X}_1 with tape S_1 after the execution halts. Subsequently, \mathfrak{X}_1 starts executing with tape T_1 and evolves to \mathfrak{X}_2 with tape S_2 . This means that when ex-machine \mathfrak{X}_1 starts executing on tape T_1 , its instructions are preserved after the halt with tape S_1 . The ex-machine evolution continues until \mathfrak{X}_{i-1} starts executing with tape T_{i-1} and evolves to ex-machine \mathfrak{X}_i with tape S_i after the execution halts. One says that the ex-machine \mathfrak{X}_0 evolves to \mathfrak{X}_i after i halts.

When \mathfrak{X}_0 evolves to \mathfrak{X}_1 , then \mathfrak{X}_1 evolves to \mathfrak{X}_2 , and so on up to \mathfrak{X}_n , then \mathfrak{X}_i is an ancestor of \mathfrak{X}_j if $0 \leq i < j \leq n$. Similarly, \mathfrak{X}_j is a descendant of \mathfrak{X}_i when $i < j$. The sequence of ex-machines $\mathfrak{X}_0 \rightarrow \mathfrak{X}_1 \rightarrow \dots \rightarrow \mathfrak{X}_n \dots$ is an evolutionary path.

3 Computing Ex-machine Languages

A class of ex-machines are evolutions of a fundamental ex-machine $\mathfrak{Z}(x)$, whose 15 instructions are listed in Definition 9. These ex-machines compute languages L that are subsets of $\{a\}^* = \{a^n : n \in \mathbb{N}\}$. a^n stands for n a's. The empty string is a^0 and $a^3 = aaa$. Set language space $\mathfrak{L} = \bigcup_{L \subset \{a\}^*} \{L\}$. Function $f : \mathbb{N} \rightarrow \{0, 1\}$ defines language L_f .

Definition 7 (Language L_f) $f : \mathbb{N} \rightarrow \{0, 1\}$ induces language $L_f = \{a^n : f(n) = 1\}$. String a^n is in L_f iff $f(n) = 1$.

Trivially, L_f is a language in \mathfrak{L} . Moreover, these functions f generate all of \mathfrak{L} .

Remark 4 (Language Space) $\mathfrak{L} = \bigcup_{f \in \{0,1\}^{\mathbb{N}}} \{L_f\}$.

Definition 8 (\mathfrak{X} Computes Language L in \mathfrak{L}) Set alphabet $A = \{\#, 0, 1, N, Y, a\}$. Let \mathfrak{X} be an ex-machine. The language L in \mathfrak{L} that \mathfrak{X} computes is defined as follows. A valid initial tape has the form $\# \# a^n \#$. The valid initial tape $\# \# \#$ represents the empty string. After \mathfrak{X} starts executing with initial tape $\# \# a^n \#$, string a^n is in \mathfrak{X} 's language if \mathfrak{X} halts with tape $\# a^n \# \ Y \#$. String a^n is not in \mathfrak{X} 's language if \mathfrak{X} halts with tape $\# a^n \# \ N \#$.

The use of special alphabet symbols Y and N —to decide whether a^n is in the language—follows [18]. For string $\# \# a^m \#$, some \mathfrak{X} could first halt with $\# a^m \# \ N \#$ and in a second execution could halt with $\# a^m \# \ Y \#$. The oscillation of halting outputs can continue indefinitely, and \mathfrak{X} 's language is not well defined per Definition 8. In this chapter, we avoid ex-machines whose halting outputs do not stabilize.

3.1 Ex-machine $\mathfrak{Z}(x)$

The purpose of Definition 9 is to show that $\mathfrak{Z}(x)$ can evolve to compute any language L_f in \mathfrak{L} ; and that evolutions of $\mathfrak{Z}(x)$ compute Turing incomputable languages on a set of Lebesgue measure 1 in language space \mathfrak{L} , where \mathfrak{L} also has measure 1.

Definition 9 (Ex-machine $\mathfrak{Z}(x)$) $A = \{\#, 0, 1, N, Y, a\}$. States $\Omega = \{0, h, n, y, t, v, w, x, 8\}$ where halting state $h = 1$ and states $n = 2, y = 3, t = 4, v = 5, w = 6, x = 7$. The initial state is always 0. For the reader's benefit, letters represent states instead of explicit numbers. State n indicates NO that the string is not in the language. State y indicates YES that the string is in the language. State x helps generate a new random bit.

$(0, \#, 8, \#, 1)$	$(8, \#, x, \#, 0)$
$(y, \#, h, Y, 0)$	$(n, \#, h, N, 0)$
$(x, \#, x, 0)$	$(x, a, t, 0)$

$(|\Omega| - 1, a, x, a, 0)$
 $(|\Omega| - 1, \#, x, \#, 0)$
 $(x, 0, v, \#, 0, (|\Omega| - 1, \#, n, \#, 1))$
 $(x, 1, w, \#, 0, (|\Omega| - 1, \#, y, \#, 1))$
 $(t, 0, w, a, 0, (|\Omega| - 1, \#, n, \#, 1))$
 $(t, 1, w, a, 0, (|\Omega| - 1, \#, y, \#, 1))$
 $(v, \#, n, \#, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$
 $(w, \#, y, \#, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$
 $(w, a, |\Omega|, a, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$

With initial state 0 and tape # #aaaa##, an execution instance of $\mathfrak{Z}(x)$ is below.

STATE	TAPE	HEAD	INSTRUCTION EXECUTED	NEW INSTRUCTION
8	# aaaa###	1	(0, #, 8, #, 1)	
x	# aaaa###	1	(8, a, x, a, 0)	(8, a, x, a, 0)
t	# 1aaa###	1	(x, a, t, 1 _r , 0)	
w	# aaaa###	1	(t, 1, w, a, 0, (\Omega - 1, #, y, #, 1))	(8, #, y, #, 1)
9	#a aaa###	2	(w, a, \Omega , a, 1, (\Omega - 1, a, \Omega , a, 1))	(8, a, 9, a, 1)
x	#a aaa###	2	(9, a, x, a, 0)	(9, a, x, a, 0)
t	#a 1aa###	2	(x, a, t, 1 _r , 0)	
w	#a aaa###	2	(t, 1, w, a, 0, (\Omega - 1, #, y, #, 1))	(9, #, y, #, 1)
10	#aa aa###	3	(w, a, \Omega , a, 1, (\Omega - 1, a, \Omega , a, 1))	(9, a, 10, a, 1)
x	#aa aa###	3	(10, a, x, a, 0)	(10, a, x, a, 0)
t	#aa 0a###	3	(x, a, t, 0 _r , 0)	
w	#aa aa###	3	(t, 0, w, a, 0, (\Omega - 1, #, n, #, 1))	(10, #, n, #, 1)
11	#aaa a###	4	(w, a, \Omega , a, 1, (\Omega - 1, a, \Omega , a, 1))	(10, a, 11, a, 1)
x	#aaa a###	4	(11, a, x, a, 0)	(11, a, x, a, 0)
t	#aaa 1###	4	(x, a, t, 1 _r , 0)	
w	#aaa a###	4	(t, 1, w, a, 0, (\Omega - 1, #, y, #, 1))	(11, #, y, #, 1)
12	#aaa ###	5	(w, a, \Omega , a, 1, (\Omega - 1, a, \Omega , a, 1))	(11, a, 12, a, 1)
x	#aaaa ###	5	(12, #, x, #, 0)	(12, #, x, #, 0)
x	#aaaa 0##	5	(x, #, x, 0 _r , 0)	
v	#aaaa ###	5	(x, 0, v, #, 0, (\Omega - 1, #, n, #, 1))	(12, #, n, #, 1)
n	#aaaa #	6	(v, #, n, #, 1, (\Omega - 1, a, \Omega , a, 1))	(12, a, 13, a, 1)
h	#aaaa# N#	6	(n, #, h, N, 0)	

This instance of $\mathfrak{Z}(x)$'s execution replaces $(8, \#, x, \#, 0)$ with $(8, \#, y, \#, 1)$. Instruction $(w, a, |\Omega|, a, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$ replaces $(8, a, x, a, 0)$ with new instruction $(8, a, 9, a, 1)$. Also, the simple meta instruction $(|\Omega| - 1, a, x, a, 0)$ temporarily added instructions $(9, a, x, a, 0)$, $(10, a, x, a, 0)$, and $(11, a, x, a, 0)$. These instructions are replaced by $(9, a, 10, a, 1)$, $(10, a, 11, a, 1)$, and $(11, a, 12, a, 1)$, respectively. Instruction $(|\Omega| - 1, \#, x, \#, 0)$ added $(12, \#, x, \#, 0)$ and instruction $(12, \#, n, \#, 1)$ replaced $(12, \#, x, \#, 0)$. Instructions $(9, \#, y, \#, 1)$, $(10, \#, n, \#, 1)$, $(11, \#, y, \#, 1)$, and $(12, a, 13, a, 1)$ are added. Five new states 9, 10, 11, 12, and 13 are added to Ω . After halting, $\Omega = \{0, h, n, y, t, v, w, x, 8, 9, 10, 11, 12, 13\}$, and the evolved ex-machine $\mathfrak{Z}(11010.x)$ has 24 instructions.

Two different instances of $\mathfrak{Z}(x)$ can evolve to two different machines and compute distinct languages according to Definition 8. After $\mathfrak{Z}(x)$ has evolved to a new machine $\mathfrak{Z}(a_0a_1 \dots a_m x)$ as a result of a prior execution with input tape # #a^m#, then for each i with $0 \leq i \leq m$, machine $\mathfrak{Z}(a_0a_1 \dots a_m x)$ always halts with the same output when presented with input tape # #aⁱ#. $\mathfrak{Z}(a_0a_1 \dots a_m x)$'s halting output stabilizes on all input strings aⁱ where $0 \leq i \leq m$. Example 4 shows this stabilization property.

Example 4 (Ex-machine $\mathfrak{Z}(1101 x)$)

```
(0, #, 8, #, 1)      (y, #, h, Y, 0)      (n, #, h, N, 0)

(x, #, x, 0)        (x, a, t, 0)
(x, 0, v, #, 0, (|\Omega| - 1, #, n, #, 1))
(x, 1, w, #, 0, (|\Omega| - 1, #, y, #, 1))

(t, 0, w, a, 0, (|\Omega| - 1, #, n, #, 1))
(t, 1, w, a, 0, (|\Omega| - 1, #, y, #, 1))

(v, #, n, #, 1, (|\Omega| - 1, a, |\Omega|, a, 1))
(w, #, y, #, 1, (|\Omega| - 1, a, |\Omega|, a, 1))
(w, a, |\Omega|, a, 1, (|\Omega| - 1, a, |\Omega|, a, 1))

(|\Omega| - 1, a, x, a, 0)
(|\Omega| - 1, #, x, #, 0)

(8, #, y, #, 1)      (8, a, 9, a, 1)
(9, #, y, #, 1)      (9, a, 10, a, 1)
(10, #, n, #, 1)     (10, a, 11, a, 1)
(11, #, y, #, 1)     (11, a, 12, a, 1)
(12, #, n, #, 1)     (12, a, 13, a, 1)
```

New instructions $(8, \#, y, \#, 1)$, $(9, \#, y, \#, 1)$, and $(11, \#, y, \#, 1)$ help $\mathfrak{Z}(11010x)$ compute that the empty strings a and aaa are in its language, respectively. Similarly, the new instructions $(10, \#, n, \#, 1)$ and $(12, \#, n, \#, 1)$ help $\mathfrak{Z}(11010x)$ compute that aa and $aaaa$ are not in its language, respectively. The 1's in $\mathfrak{Z}(11010x)$'s name indicate that the empty strings a and aaa are in its language. The 0's indicate that strings aa and $aaaa$ are not in its language. Symbol x indicates that $\mathfrak{Z}(11010x)$ has not yet determined for $n \geq 5$ whether strings a^n are in $\mathfrak{Z}(11010x)$'s language.

Starting at state 0, $\mathfrak{Z}(11010x)$ computes that the empty string is in its language

STATE	TAPE	HEAD	INSTRUCTION
8	## ###	1	(0, #, 8, #, 1)
y	### #	2	(8, #, y, #, 1)
h	### Y#	2	(y, #, h, Y, 0)

Starting at state 0, $\mathfrak{Z}(11010x)$ computes that string aa is not in its language.

STATE	TAPE	HEAD	INSTRUCTION
8	## aa###	1	(0, #, 8, #, 1)
9	##a a###	2	(8, a, 9, a, 1)
10	##aa ###	3	(9, a, 10, a, 1)
n	##aa# #	4	(10, #, n, #, 1)
h	##aa# N#	4	(n, #, h, N, 0)

Similarly, starting at state 0, $\mathfrak{Z}(11010x)$ computes that a and aaa are in its language and $\mathfrak{Z}(11010x)$ computes that $aaaa$ is not in its language. For each of these executions, no new states are added and no instructions are added or replaced. Thus, for all subsequent executions, $\mathfrak{Z}(11010x)$ computes that the empty strings a and aaa are in its language, and strings aa and $aaaa$ are not.

Starting at state 0, below is an execution of $\mathfrak{Z}(11010x)$ on input tape # #aaaaa##.

STATE	TAPE	HEAD	INSTRUCTION EXECUTED	NEW INSTRUCTION
8	# aaaaaa##	1	(0, #, 8, #, 1)	
9	#a aaaaa##	2	(8, a, 9, a, 1)	
10	#aa aaaa##	3	(9, a, 10, a, 1)	
11	#aaa aaa##	4	(10, a, 11, a, 1)	
12	#aaaa aa##	5	(11, a, 12, a, 1)	
13	#aaaaa a##	6	(12, a, 13, a, 1)	
x	#aaaaa a##	6	(13, a, x, a, 0)	
t	#aaaaa 0##	6	(x, a, t, 0 _r , 0)	
w	#aaaaa a##	6	(t, 0, w, a, 0, (Ω - 1, #, n, #, 1))	(13, #, n, #, 1)
14	#aaaaaa ##	7	(w, a, Ω , a, 1, (Ω - 1, a, Ω , a, 1))	(13, a, 14, a, 1)
x	#aaaaaa ##	7	(14, #, x, #, 0)	(14, #, x, #, 0)
x	#aaaaaa 1#	7	(x, #, x, 1 _r , 0)	
w	#aaaaaa ##	7	(x, 1, w, #, 0, (Ω - 1, #, y, #, 1))	(14, #, y, #, 1)
y	#aaaaaa# #	8	(w, #, y, #, 1, (Ω - 1, a, Ω , a, 1))	(14, a, 15, a, 1)
h	#aaaaaa# Y	8	(y, #, h, Y, 0)	

$\mathfrak{Z}(11010x)$ evolves to $\mathfrak{Z}(1101001x)$. The first random instruction $(x, a, t, 0)$ measures a 0, so it executes as $(x, a, t, 0_r, 0)$. Instruction $(13, \#, n, \#, 1)$ is added due to the random 0 bit; in all subsequent executions of $\mathfrak{Z}(1101001x)$, string a^5 is not in $\mathfrak{Z}(1101001x)$'s language. The second random instruction $(x, \#, x, 0)$ measures a 1 and executes as $(x, \#, x, 1_r, 0)$. Instruction $(14, \#, y, \#, 1)$ is added. In all subsequent executions, string a^6 is in $\mathfrak{Z}(1101001x)$'s language.

Definition 10 specifies $\mathfrak{Z}(a_0a_1 \dots a_mx)$ and covers $\mathfrak{Z}(11010x)$'s execution.

Definition 10 (Ex-machine $\mathfrak{Z}(a_0a_1 \dots a_mx)$) Let $m \in \mathbb{N}$. Set $\Omega = \{0, h, n, y, t, v, w, x, 8, 9, 10, \dots, m+8, m+9\}$. For $0 \leq i \leq m, a_i$ is 0 or 1. In $\mathfrak{Z}(a_0a_1 \dots a_mx)$'s instructions, symbol $b_8 = y$ if $a_0 = 1$, else $b_8 = n$ if $a_0 = 0$; symbol $b_9 = y$ if $a_1 = 1$, else $b_9 = n$ if $a_1 = 0$; and so on until the second to the last instruction $(m+8, \#, b_{m+8}, \#, 1), b_{m+8} = y$ if $a_m = 1$, else $b_{m+8} = n$ if $a_m = 0$.

- $(0, \#, 8, \#, 1)$ $(y, \#, h, Y, 0)$ $(n, \#, h, N, 0)$
- $(x, \#, x, 0)$
- $(x, a, t, 0)$
- $(|\Omega| - 1, a, x, a, 0)$
- $(|\Omega| - 1, \#, x, \#, 0)$
- $(x, 0, v, \#, 0, (|\Omega| - 1, \#, n, \#, 1))$
- $(x, 1, w, \#, 0, (|\Omega| - 1, \#, y, \#, 1))$
- $(t, 0, w, a, 0, (|\Omega| - 1, \#, n, \#, 1))$
- $(t, 1, w, a, 0, (|\Omega| - 1, \#, y, \#, 1))$
- $(v, \#, n, \#, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$
- $(w, \#, y, \#, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$
- $(w, a, |\Omega|, a, 1, (|\Omega| - 1, a, |\Omega|, a, 1))$
- $(8, \#, b_8, \#, 1)$ $(8, a, 9, a, 1)$ $(9, \#, b_9, \#, 1)$ $(9, a, 10, a, 1)$
- $(10, \#, b_{10}, \#, 1)$ $(10, a, 11, a, 1)$. . . $(i+8, \#, b_{i+8}, \#, 1)$ $(i+8, a, i+9, a, 1)$. . .
- $(m+7, \#, b_{m+7}, \#, 1)$ $(m+7, a, m+8, a, 1)$ $(m+8, \#, b_{m+8}, \#, 1)$ $(m+8, a, m+9, a, 1)$

Lemma 1 *If i satisfies $0 \leq i \leq m$, string a^i is in $\mathfrak{Z}(a_0a_1 \dots a_m x)$'s language if $a_i = 1$, and string a^i is not in $\mathfrak{Z}(a_0a_1 \dots a_m x)$'s language if $a_i = 0$. If $n > m$, it has not yet been determined whether a^n is in $\mathfrak{Z}(a_0a_1 \dots a_m x)$'s language or not in its language.*

Proof When $0 \leq i \leq m$, the first consequence follows immediately from the definition of a^i being in $\mathfrak{Z}(a_0a_1 \dots a_m x)$'s language and from Definition 10. In instruction $(i + 8, \#, b_{i+8}, \#, 1)$, the state value of b_{i+8} is γ if $a_i = 1$ and b_{i+8} is n if $a_i = 0$.

For the indeterminacy of strings a^n when $n > m$, $\mathfrak{Z}(a_0 \dots a_m x)$ executes its last instruction $(m + 8, a, m + 9, a, 1)$ when scanning the m th a in a^n . For each a to the right of $\#a^m$ on the tape, $\mathfrak{Z}(a_0 \dots a_m x)$ executes random instruction $(x, a, t, 0)$.

If $(x, a, t, 0)$ measures 0, then meta instructions $(t, 0, w, a, 0, (|\Omega|-1, \#, n, \#, 1))$ and $(w, a, |\Omega|, a, 1, (|\Omega|-1, a, |\Omega|, a, 1))$ execute. Otherwise, $(x, a, t, 0)$ measures 1, so $(t, 1, w, a, 0, (|\Omega|-1, \#, \gamma, \#, 1))$ and $(w, a, |\Omega|, a, 1, (|\Omega|-1, a, |\Omega|, a, 1))$ execute. If the next alphabet symbol to the right is an a , then a new standard instruction executes, derived from an instantiation of $(|\Omega|-1, a, x, a, 0)$. When the tape head scans the last a in a^n , a new standard instruction executes, derived from $(|\Omega|-1, \#, x, \#, 0)$.

For each a to the right of $\#a^m$ on the tape, the execution of random instruction $(x, a, t, 0)$ determines whether string a^{m+k} , such that $1 \leq k \leq n - m$, is in $\mathfrak{Z}(a_0a_1 \dots a_n x)$'s language. After the execution of $(|\Omega|-1, \#, x, \#, 0)$, the tape head is scanning a blank symbol, so the random instruction $(x, \#, x, 0)$ is executed. If the random source generates 0, the meta instructions $(x, 0, v, \#, 0, (|\Omega|-1, \#, n, \#, 1))$ and $(v, \#, n, \#, 1, (|\Omega|-1, a, |\Omega|, a, 1))$ execute. Then, instruction $(n, \#, h, N, 0)$ executes last, which indicates that a^n is not in $\mathfrak{Z}(a_0a_1 \dots a_n x)$'s language. If the execution of $(x, \#, x, 0)$ measures 1, the instructions $(x, 1, w, \#, 0, (|\Omega|-1, \#, \gamma, \#, 1))$ and $(w, \#, \gamma, \#, 1, (|\Omega|-1, a, |\Omega|, a, 1))$ execute. Then, instruction $(\gamma, \#, h, Y, 0)$ executes last, which indicates that a^n is in $\mathfrak{Z}(a_0a_1 \dots a_n x)$'s language. During the execution of the instructions, for each a on the tape to the right of $\#a^m$, $\mathfrak{Z}(a_0a_1 \dots a_m x)$ evolves to $\mathfrak{Z}(a_0a_1 \dots a_n x)$ according to the instructions, specified by Definition 10, where one substitutes n for m .

3.2 Some Turing Incomputable Properties of $\mathfrak{Z}(x)$

When the measurements in $\mathfrak{Z}(x)$'s two random instructions satisfy both axioms, all 2^n finite paths of length n in the infinite binary tree of Fig. 1 are equally likely. The 1-to-1 correspondence between $f : \mathbb{N} \rightarrow \{0, 1\}$ and an infinite downward path (red) in the binary tree helps show that $\mathfrak{Z}(x)$ can evolve to compute any language L_f in \mathfrak{L} .

Consider $\mathfrak{Z}(x)$ and all $\mathfrak{Z}(a_0 \dots a_m x)$ for each $m \in \mathbb{N}$ and $a_0 \dots a_m$ in $\{0, 1\}^{m+1}$.

Theorem 1 *Each language L_f in \mathfrak{L} can be computed by the evolving sequence of ex-machines $\mathfrak{Z}(x)$, $\mathfrak{Z}(f(0)x)$, $\mathfrak{Z}(f(0)f(1)x)$, \dots , $\mathfrak{Z}(f(0)f(1) \dots f(n)x)$, \dots*

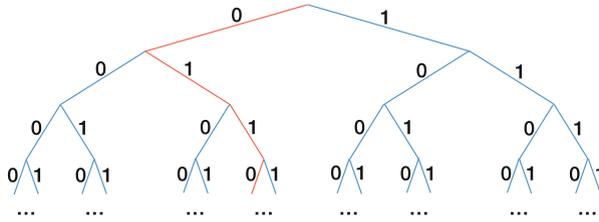


Fig. 1 Infinite binary tree. A graphical representation of $\{0, 1\}^{\mathbb{N}}$

Proof Apply Definitions 9 and 10 and Lemma 1.

Corollary 1 For any $f : \mathbb{N} \rightarrow \{0, 1\}$ and any n , the evolving sequence $\mathfrak{Z}(f(0)x), \dots, \mathfrak{Z}(f(0)f(1)\dots f(n)f(n+1)x), \dots$ computes language L_f .

Corollary 2 For each n , the evolution of ex-machines $\mathfrak{Z}(x), \mathfrak{Z}(f(0)x), \mathfrak{Z}(f(0)f(1)x), \dots, \mathfrak{Z}(f(0)f(1)\dots f(n)x)$ have cumulatively used only a finite amount of tape, finite number of states, finite number of instructions, and finite number of instruction executions, and only a finite amount of quantum information is measured by the random instructions.

Proof Remark 3 and Definitions 5 and 10 imply finite computational resources.

Theorem 2 and Corollary 3 come from the following intuition. A set X is countable if there exists a bijection between X and \mathbb{N} . \mathfrak{L} is uncountable, so most languages L_f in \mathfrak{L} are Turing incomputable. Since each L_f is equally likely of being computed by $\mathfrak{Z}(x)$, most languages computed by $\mathfrak{Z}(x)$'s evolution are Turing incomputable.

For each $n \in \mathbb{N}$, define language tree $\mathfrak{L}(a_0 \dots a_n) = \{L_f : f \in \{0, 1\}^{\mathbb{N}} \text{ and } f(i) = a_i \text{ for } i \text{ such that } 0 \leq i \leq n\}$. Define subtree $\mathfrak{S}(a_0 \dots a_n) = \{f \in \{0, 1\}^{\mathbb{N}} : f(i) = a_i \text{ such that } 0 \leq i \leq n\}$. Let Ψ be this 1-to-1 correspondence: $\mathfrak{L} \xrightarrow{\Psi} \{0, 1\}^{\mathbb{N}}$ and $\mathfrak{L}(a_0 \dots a_n) \xrightarrow{\Psi} \mathfrak{S}(a_0 \dots a_n)$. Since random axioms 1 and 2 hold, each finite path $f(0)f(1)\dots f(n)$ is equally likely. There are 2^{n+1} of these paths, so each path has probability $2^{-(n+1)}$. The uniform probabilities on finite strings of the same length extend to Lebesgue [9, 22] measure μ on probability space $\{0, 1\}^{\mathbb{N}}$. Subtree $\mathfrak{S}(a_0 \dots a_n)$ has measure $2^{-(n+1)}$, where $\mu(\mathfrak{S}(a_0 \dots a_n)) = 2^{-(n+1)}$ and $\mu(\{0, 1\}^{\mathbb{N}}) = 1$. Via Ψ , μ induces uniform probability measure ν on \mathfrak{L} , where $\nu(\mathfrak{L}(a_0 \dots a_n)) = 2^{-(n+1)}$ and $\nu(\mathfrak{L}) = 1$.

Theorem 2 The Turing incomputable languages L_f have measure 1 in (ν, \mathfrak{L}) .

Proof The Turing computable functions $f : \mathbb{N} \rightarrow \{0, 1\}$ are countable. Via the Ψ correspondence, the Turing computable languages L_f have ν -measure 0 in \mathfrak{L} .

Corollary 3 For all $a_0 \dots a_m$ in $\{0, 1\}^{m+1}$, $\mathfrak{Z}(a_0 \dots a_m x)$ is not a Turing machine.

Proof $\mathfrak{Z}(x)$ can evolve to compute Turing incomputable languages on a set of ν -measure 1 in \mathfrak{L} . $\mathfrak{Z}(a_0 \dots a_m x)$ can evolve to compute Turing incomputable

languages on a set of ν -measure $2^{-(m+1)}$ in \mathfrak{L} . Each Turing machine only computes one language, so the measure of all Turing computable languages is 0 in \mathfrak{L} .

4 An Ex-machine Halting Problem

In [23], Turing posed the question, does there exist a Turing machine \mathfrak{D} that can determine for any given Turing machine M and finite tape T whether M 's execution on tape T eventually halts? Turing proved that no Turing machine could solve this problem. His halting problem can be extended to ex-machines. Does there exist an ex-machine $\mathfrak{X}(x)$ such that for any given Turing machine M , then $\mathfrak{X}(x)$ can sometimes compute whether M 's execution on finite initial tape T will eventually halt? In order for this question to be well-posed, the phrase *can sometimes compute whether* must be defined.

From the universal Turing machine / enumeration theorem [21], there is a Turing computable enumeration $\mathfrak{E} : \mathbb{N} \rightarrow \{\text{Turing machines } M\} \times \{\text{Each initial state of } M\}$ of every Turing machine. Similar to ex-machines, for each machine M , the set $\{\text{Each initial state of } M\}$ is realized as a finite subset $\{0, \dots, n - 1\}$ of \mathbb{N} . Since $\mathfrak{E}(n)$ is an ordered pair, the phrase "Turing machine $\mathfrak{E}(n)$ " refers to the first coordinate of $\mathfrak{E}(n)$. The "initial state $\mathfrak{E}(n)$ " refers to the second coordinate of $\mathfrak{E}(n)$. Turing's halting problem is equivalent to the blank-tape halting problem [19]. The blank-tape halting problem translates to: for each Turing machine $\mathfrak{E}(n)$, does $\mathfrak{E}(n)$ halt when $\mathfrak{E}(n)$ begins executing with a blank initial tape and initial state $\mathfrak{E}(n)$?

Lemma 1 implies that the same initial ex-machine can evolve to two different ex-machines; these two ex-machines will never compute the same language no matter what descendants they evolve to. For example, $\mathfrak{Z}(0x)$ and $\mathfrak{Z}(1x)$ can never compute the same language in \mathfrak{L} . Hence, *sometimes* means that for each n , there exists an evolution of $\mathfrak{X}(x)$ to $\mathfrak{X}(a_0x)$, then to $\mathfrak{X}(a_0a_1x)$, and so on up to $\mathfrak{X}(a_0a_1 \dots a_n x) \dots$, where for each i with $0 \leq i \leq n$, then $\mathfrak{X}(a_0a_1 \dots a_n x)$ correctly computes whether Turing machine $\mathfrak{E}(n)$ halts or does not halt. The word *computes* means that $\mathfrak{X}(a_0a_1 \dots a_i x)$ halts after a finite number of instructions executed, and the halting output written by $\mathfrak{X}(a_0a_1 \dots a_i x)$ on the tape indicates whether machine $\mathfrak{E}(n)$ halts. For example, if the input tape is $\# \#a^i\#$, then enumeration machine $M_{\mathfrak{E}}$ writes the representation of $\mathfrak{E}(i)$ on the tape, and then $\mathfrak{X}(a_0a_1 \dots a_m x)$ with $m \geq i$ halts with $\# \Upsilon\#$ written to the right of the representation for machine $\mathfrak{E}(i)$. Alternatively, $\mathfrak{X}(a_0a_1 \dots a_m x)$ with $m \geq i$ halts with $\# \text{N}\#$ written to the right of the representation for machine $\mathfrak{E}(i)$. The word *correctly* means that ex-machine $\mathfrak{X}(a_0a_1 \dots a_m x)$ halts with $\# \Upsilon\#$ written on the tape if machine $\mathfrak{E}(i)$ halts and ex-machine $\mathfrak{X}(a_0a_1 \dots a_m x)$ halts with $\# \text{N}\#$ written on the tape if machine $\mathfrak{E}(i)$ does not halt.

Next, the ex-machine halting problem is transformed so that the results from Sect. 3 can be applied. Choose alphabet $\mathfrak{A} = \{\#, 0, 1, a, A, B, M, N, S, X, Y\}$.

As before, identify the set of Turing machine states Ω as a finite subset of \mathbb{N} . Let $M_{\mathfrak{E}}$ be the Turing machine that computes a Turing computable enumeration ⁴ as $\mathfrak{E}_a : \mathbb{N} \rightarrow \{\mathfrak{A}\}^* \times \mathbb{N}$, where the tape $\# \# \mathfrak{a}^n \#$ represents natural number n . Each $\mathfrak{E}_a(n)$ is an ordered pair where the first coordinate is the Turing machine and the second coordinate is an initial state chosen from $\mathfrak{E}_a(n)$'s states. Define the *halting function* $h_{\mathfrak{E}_a} : \mathbb{N} \rightarrow \{0, 1\}$ such that for each n , set $h_{\mathfrak{E}_a}(n) = 1$, whenever $\mathfrak{E}_a(n)$ halts. Otherwise, set $h_{\mathfrak{E}_a}(n) = 0$, if $\mathfrak{E}_a(n)$ with blank initial tape and initial state $\mathfrak{E}_a(n)$ does not halt. Function $h_{\mathfrak{E}_a}(n)$ is well defined because for each $n \in \mathbb{N}$, with blank initial tape and initial state $\mathfrak{E}_a(n)$, Turing machine $\mathfrak{E}_a(n)$ either halts or does not halt. Via function $h_{\mathfrak{E}_a}(n)$ and Definition 7, define *halting language* $L_{h_{\mathfrak{E}_a}}$.

Theorem 3 *There exists an evolutionary path for ex-machine $\mathfrak{Z}(x)$ that computes halting language $L_{h_{\mathfrak{E}_a}}$; namely, $\mathfrak{Z}(h_{\mathfrak{E}_a}(0) x) \rightarrow \mathfrak{Z}(h_{\mathfrak{E}_a}(0) h_{\mathfrak{E}_a}(1) x) \rightarrow \dots \mathfrak{Z}(h_{\mathfrak{E}_a}(0) h_{\mathfrak{E}_a}(1) \dots h_{\mathfrak{E}_a}(m) x) \dots$*

Proof Apply the mathematical developments in the previous three paragraphs, using halting function $h_{\mathfrak{E}_a}$, language $L_{h_{\mathfrak{E}_a}}$, and Theorem 1.

Theorem 3 implies that a proof by contradiction for Turing's halting problem does not hold for ex-machines: the existence of path $\mathfrak{Z}(h_{\mathfrak{E}_a}(0) x) \rightarrow \mathfrak{Z}(h_{\mathfrak{E}_a}(0) h_{\mathfrak{E}_a}(1) x) \rightarrow \dots \mathfrak{Z}(h_{\mathfrak{E}_a}(0) h_{\mathfrak{E}_a}(1) \dots h_{\mathfrak{E}_a}(m) x) \dots$ circumvents the contradiction. From an information-theoretic perspective, almost every (w.r.t. to μ on $\{0, 1\}^{\mathbb{N}}$) evolutionary path $\mathfrak{Z}(f(0) x) \rightarrow \mathfrak{Z}(f(0)f(1) x) \rightarrow \dots \mathfrak{Z}(f(0)f(1) \dots f(n) x) \dots$ avoids the contradiction in Chaitin's information-theoretic proof [5] that the halting problem for Turing machines is unsolvable. The contradiction depends upon the following: the minimum number of bits needed to represent a Turing machine stays constant. In contrast, there is a set $\mathfrak{F} \subset \{0, 1\}^{\mathbb{N}}$ with $\mu(\mathfrak{F}) = 1$ such that for all $f \in \mathfrak{F}$, the minimum number of bits needed to represent $\mathfrak{Z}(f(0)f(1) \dots f(n) x)$ increases without bound as n increases.

5 A Research Direction

Theorem 3 and information-theoretic analysis both show that a proof by contradiction of the unsolvability of Turing's halting problem does not apply to ex-machines. This capability suggests that novel self-modification procedures, cleverly integrated with randomness, should be explored to help enhance theorem proving [2, 6] and constructive type systems that use conservative workarounds [20] to avoid the halting problem.

⁴Chapter 7 of [19] provides explicit details of encoding quintuples with a particular universal Turing machine. Alphabet \mathfrak{A} was selected to be compatible with this encoding. A careful study of chapter 7 provides a clear path of how $M_{\mathfrak{E}}$'s instructions can be specified to implement \mathfrak{E}_a .

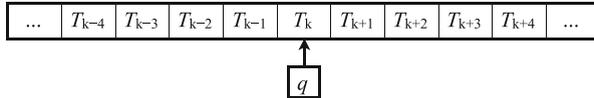


Fig. 2 Machine configuration (q, k, T) before executing a standard instruction

6 Conclusion

We showed that ex-machines can compute Turing incomputable languages and that ex-machines are not limited by the halting problem for Turing machines. The language computed by an ex-machine reflects its computational capabilities. The problem of determining program correctness for a digital computer program is unsolvable by a Turing machine. The detection of an infinite loop in a computer program (i.e., a case of program correctness) can be reduced to Turing’s halting problem. For these reasons, it is important to understand how far methods of evaluating program correctness for digital computer programs can be extended with randomness and advanced self-modification procedures.

Appendix: A Turing Machine Is an Autonomous Dynamical System

Fix a Turing machine M . Transformation ϕ maps a machine configuration to a point in the complex plane \mathbb{C} ; ϕ also maps each of M ’s instructions to a finite number $|A|$ of unique affine functions each with a distinct domain. These affine functions can be extended to a function F on a bounded region W in \mathbb{C} , containing these domains and a disjoint bounded set, called the halting attractor. Via ϕ , one computational step of M corresponds to one iteration of the discrete autonomous dynamical system (F, W) .

Let machine states $\mathbf{Q} = \{q_1, \dots, q_m\}$. Let alphabet $A = \{a_1, \dots, a_n\}$, where a_1 is the blank symbol. Halt state h is a special state that is not in \mathbf{Q} . Function $\eta : \mathbf{Q} \times A \rightarrow \mathbf{Q} \cup \{h\} \times A \times \{-1, +1\}$ is the machine M ’s program. A single instruction is $\eta(q, a) = (r, b, x)$, where $q \in \mathbf{Q}$, $r \in \mathbf{Q} \cup \{h\}$, $a, b \in A$, and $x \in \{-1, +1\}$. Set $B = |A| + |\mathbf{Q}| + 1$. Define symbol value function $v : \{h\} \cup \mathbf{Q} \cup A \rightarrow \mathbb{N}$ as $v(a_1) = 0, \dots, v(a_i) = i - 1, \dots, v(a_n) = |A| - 1, v(h) = |A|, v(q_1) = |A| + 1, \dots, v(q_i) = |A| + i, \dots, v(q_m) = |A| + |\mathbf{Q}|$.

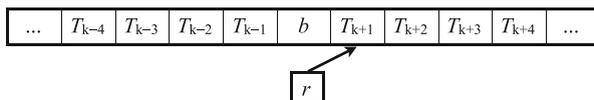


Fig. 3 Machine configuration after executing instruction $\eta(q, T_k) = (r, b, +1)$

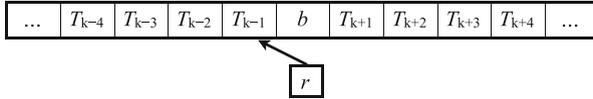


Fig. 4 Machine configuration after executing instruction $\eta(q, T_k) = (r, b, -1)$

$T : \mathbb{Z} \rightarrow A$ is the tape and is finite. T_k is the alphabet symbol in tape square k . Machine configuration (q, k, T) lies in $\mathbf{Q} \times \mathbb{Z} \times A^{\mathbb{Z}}$ and maps to the complex number:

$$\phi(q, k, T) = |A|v(T_k) + \sum_{j=0}^{\infty} v(T_{k+j+1})|A|^{-j} + \left(Bv(q) + \sum_{j=0}^{\infty} v(T_{k-j-1})|A|^{-j} \right) i. \tag{1}$$

In Eq. 1, the infinite series in both the real and imaginary parts sums to rational numbers because the initial tape squares contain a finite number of non-blank symbols.

Next, we define how ϕ maps each instruction in program η to a finite set of affine functions. When instruction $\eta(q, T_k) = (r, b, +1)$ executes, state q moves to state r , symbol b replaces T_k on tape square k , and the head moves to tape square $k + 1$.

The right affine functions corresponding to instruction $\eta(q, T_k) = (r, b, +1)$ are of the form $f(x + yi) = f_1(x) + f_2(y) i$, where $f_1(x) = |A|x + m$ and $f_2(y) = \frac{1}{|A|}y + n$. Using Eq. 1 and Fig. 3 to solve for m and n , ϕ maps instruction $\eta(q, T_k) = (r, \alpha, +1)$ to the affine function $f(x + yi) = f_1(x) + f_2(y) i$, where

$$f_1(x) = |A|x + (|A| - 1)v(T_{k+1}) - |A|^2v(T_k) \tag{2}$$

$$f_2(y) = \frac{1}{|A|}y + Bv(r) + v(b) - \frac{B}{|A|}v(q). \tag{3}$$

For each of the $|A|$ distinct values $v(T_{k+1})$ in f_1 , f is a different affine function. Thus, there are $|A|$ distinct affine functions that correspond to instruction $\eta(q, T_k) = (r, b, +1)$. The domain of each right affine function is $U_{j,k} = \{x + yi \in \mathbb{C} : j \leq x < j + 1 \text{ and } k \leq y < k + |A|\}$, where $j = |A|v(T_k) + v(T_{k+1})$ and $k = Bv(q)$.

When $\eta(q, T_k) = (r, b, -1)$ executes, state q moves to state r , symbol b replaces T_k on square k , and the head moves to tape square $k - 1$.

From Eq. 1 and Fig. 4, ϕ maps instruction $\eta(q, T_k) = (r, b, -1)$ to affine function $g(x + yi) = g_1(x) + g_2(y) i$, where

$$g_1(x) = \frac{1}{|A|}x + |A|v(T_{k-1}) + v(b) - v(T_k) \tag{4}$$

$$g_2(y) = |A|y + Bv(r) - |A|Bv(q) - |A|v(T_{k-1}). \tag{5}$$

For each of the $|A|$ distinct values $v(T_{k-1})$ in g_1 and g_2 , g is a different affine function. Thus, there are $|A|$ distinct left affine functions that correspond

to instruction $\eta(q, T_k) = (r, b, -1)$. The domain of each left affine function is $V_{j,k} = \{x + yi \in \mathbb{C} : j \leq x < j + |A| \text{ and } k \leq y < k + 1\}$, where $j = |A|v(T_k)$ and $k = Bv(q) + v(T_{k-1})$.

Define *halting attractor* $H = \{x + yi \in \mathbb{C} : 0 \leq x < |A|^2 \text{ and } B|A| \leq y \leq (B + 1)|A|\}$. The points in \mathbb{C} that correspond to halting configurations (h, k, T) are called *halting points*. Using elementary algebra and simple geometric series calculations, one can verify that the halting points are a subset of H . Define *halting map* $\mathfrak{h} : H \rightarrow H$, where $\mathfrak{h}(x + yi) = x + yi$ on H . Every point in the halting attractor is a fixed point of \mathfrak{h} . Moreover, the intersection of each affine function's domain and H is empty. This implies that \mathfrak{h} and all left and right affine functions corresponding to η 's instructions can be extended to a function F on domain W that contains H and all domains $U_{j,k}$ and $V_{j,k}$.

Overall, the ϕ correspondence transforms Turing's halting problem to a discrete autonomous dynamical systems problem in \mathbb{C} . If machine configuration (q, k, T) halts after n computational steps, then the orbit of $\phi(q, k, T)$ exits one of the domains $U_{j,k}$ or $V_{j,k}$ on the n th iteration and enters the halting attractor H . If machine configuration (r, j, S) never halts, then the orbit of $\phi(r, j, S)$ never reaches the halting attractor.

References

1. H. Abelson, G.J. Sussman, J. Sussman, *Structure and Interpretation of Computer Programs*, 2nd edn. (MIT Press, Cambridge, 1996)
2. Y. Bertot, P. Castéran, *Interactive Theorem Proving & Program Development* (Springer, Berlin, 2004)
3. M. Blum, On the size of machines. *Inf. Control* **11**, 257–265 (1967)
4. C. Calude, *Information and Randomness* (Springer, Berlin, 2002), pp. 362–363
5. G. Chaitin. *Information, Randomness, and Incompleteness* (World Scientific, Singapore, 1987)
6. T. Coquand, G. Huet, Calculus of constructions. *Inf. Comput.* **76**, 95–120 (1988)
7. K. de Leeuw, E.F. Moore, C.E. Shannon, N. Shapiro, Computability by probabilistic machines, in ed. by Shannon & McCarthy *Automata Studies* (Princeton University Press, Princeton, 1956), pp. 183–212
8. G. Etesi, I. Nemeti, Non-Turing computations via Malament-Hogarth spacetimes. *Int. J. Theoret. Phys.* **41**(2), 341–370 (2002)
9. W. Feller, *Introduction to Probability Theory and Its Applications*, vol. 1 (Wiley, Hoboken, 1957), vol. 2 (1966)
10. M.S. Fiske, *Non-autonomous Dynamical Systems Applicable to Neural Computation* (Northwestern University, Evanston, 1996)
11. M.S. Fiske, Turing incomputable computation. *Turing-100. The Alan Turing Centenary. EasyChair* **10**, 66–91 (2012). <https://doi.org/10.29007/x5g2>
12. M.S. Fiske, Quantum random self-modifiable computation. *Logic Colloquium 2019. Prague, Czech Republic, August 11–16. Bull. Symb. Logic.* **25**(4), 510–511 (2019). <https://doi.org/10.1017/bsl.2019.56>
13. H. Godfroy, J.Y. Marion, *Abstract Self Modifying Machines* (HAL CCSD, Lyon, 2016)
14. M. Herrero-Collantes, J.C. Garcia-Escartin, Quantum random number generators. *Rev. Modern Phys.* **89**(1), 015004, (2017). <https://arxiv.org/abs/1604.03304>

15. D. Hilbert, Mathematische probleme. Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematische-Physikalische Klasse **3**, 253–297 (1900)
16. M. Hogarth, Non-turing computers and Non-turing computability, in *Proceedings of the Biennial Meeting of the Philosophy of Science Assoc.*, vol. 1 (University of Chicago, Chicago, 1994), pp. 126–138
17. T. Kieu, Quantum algorithm for Hilbert’s tenth problem. *Int. J. Theoret. Phys.* **42**, pp. 1461–1478 (2003)
18. H.R. Lewis, C. Papadimitriou, *Elements of the Theory of Computation* (Prentice-Hall, Upper Saddle River, 1981)
19. M. Minsky, *Computation: Finite and Infinite Machines* (Prentice-Hall, Upper Saddle River, 1967), pp. 132–155
20. B. Pierce, *Types and Programming Languages* (MIT Press, Cambridge, 2002), pp. 99–100
21. H. Rogers. *Theory of Recursive Functions and Effective Computability* (MIT Press, Cambridge, 1987)
22. H.L. Royden, *Real Analysis* (Prentice-Hall, Upper Saddle River, 1988)
23. A.M. Turing, On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Math. Soc. Series 2.* **42** (3, 4), 230–265 (1936)
24. A.M. Turing, System of logic based on ordinals. *Proc. London Math. Soc. Series 2.* **45**, 161–228 (1939)

ECM Factorization with QRT Maps



Andrew N. W. Hone

1 Introduction

Elliptic curves are a fundamental tool in modern cryptography. The abelian group structure on an elliptic curve makes it suitable for versions of Diffie–Hellman key exchange and ElGamal key encryption, as well as providing techniques for primality testing and integer factorization, among many other applications relevant to network security [4, 22, 32, 36]. In this chapter, we consider an approach to integer factorization using elliptic curves.

The elliptic curve method (ECM) due to Lenstra [24] is one of the most effective methods known for finding medium-sized prime factors of large integers, in contrast to trial division, Pollard’s rho method, or the $p - 1$ method, which quickly find small factors, or sieve methods, which are capable of finding very large prime factors. For factoring an integer N , the basic idea of the ECM is to pick (at random) an elliptic curve E and a point $P \in E$, then compute the scalar multiple $sP = P + \dots + P$ (s times) in the group law of the curve, using arithmetic in the ring $\mathbb{Z}/N\mathbb{Z}$, take a rational function f on E with a pole at the point O corresponding to the identity in the group E , and evaluate $f(sP)$ for some s chosen as the largest prime power less than some fixed bound B_1 or as the product of all such prime powers. For certain choices of E and P , this computation may lead to an attempt to divide by a non-unit in the ring, resulting in a factor of N being found.

This work begun on leave at School of Mathematics & Statistics, University of New South Wales, NSW 2052, Australia.

A. N. W. Hone (✉)

School of Mathematics, Statistics & Actuarial Science, University of Kent, Canterbury, UK
e-mail: A.N.W.Hone@kent.ac.uk

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_28

395

To be more precise, traditionally, one starts with a Weierstrass cubic defined over \mathbb{Q} , which can be taken with integer coefficients as

$$y^2 = x^3 + Ax + B, \quad A, B \in \mathbb{Z},$$

so that arithmetic mod N corresponds to working with the pseudocurve (or group scheme) $E(\mathbb{Z}/N\mathbb{Z})$ consisting of all $(x, y) \in (\mathbb{Z}/N\mathbb{Z})^2$ that satisfy the cubic equation together with O , the point at infinity; but, when N is composite, the group addition $P_1 + P_2$ is not defined for all pairs of points $P_1, P_2 \in E(\mathbb{Z}/N\mathbb{Z})$. Typically, f is taken to be the coordinate function x , and the method is successful if computing the scalar multiple sP leads to an x -coordinate with a denominator D which is not a unit in $\mathbb{Z}/N\mathbb{Z}$, such that $\gcd(D, N) > 1$ is a non-trivial factor of N . When this fortunate occurrence arises, it indicates that there is a prime factor $p|N$ for which $sP = O$ in the group law of the bona fide elliptic curve $E(\mathbb{F}_p)$, which is guaranteed if s is a multiple of the order $\#E(\mathbb{F}_p)$.

The original description of the ECM was based on computations with affine coordinates for a Weierstrass cubic; computing the scalar multiple sP is now known as “stage 1” of the ECM, and there is a further “stage 2”, due to Brent, involving computing multiples ℓsP for small primes ℓ less than some bound $B_2 > B_1$, but here we only focus on stage 1. Improvements in efficiency can be made by using various types of projective coordinates and/or Montgomery curves (see chapter 7 in [4]). However, all of these approaches share an inconvenient feature of the addition law for $P_1 + P_2$ on a Weierstrass cubic, namely that the formulae for $P_2 = \pm P_1$ or $P_2 = O$ are different from the generic case.

An important new development was the proposal of Bernstein and Lange [1] to consider a different model for E , namely the Edwards curve [6]

$$E_d : \quad x^2 + y^2 = 1 + dx^2y^2 \tag{1}$$

(d is a parameter), for which the addition law

$$(x_1, y_1) + (x_2, y_2) = \left(\frac{x_1y_2 + y_1x_2}{1 + dx_1x_2y_1y_2}, \frac{y_1y_2 - x_1x_2}{1 - dx_1x_2y_1y_2} \right) \tag{2}$$

has the advantage that it is also valid for a generic pair of points $P_1, P_2 \in E_d$, even when $P_1 = P_2$, so it can be used for doubling (following [1], we have used a rescaled curve compared with the original version in [6]). The fact that the addition law (2) on E_d is unified in this sense is implicit in the classical addition formula for the Jacobi sine function (see chapter XXII in [35], or chapter 22 in [28]), for we have been

$$\operatorname{sn}(z + w) = \frac{\operatorname{sn}(z)\operatorname{cd}(w) + \operatorname{cd}(z)\operatorname{sn}(w)}{1 + k^2\operatorname{sn}(z)\operatorname{sn}(w)\operatorname{cd}(z)\operatorname{cd}(w)},$$

$$\operatorname{cd}(z + w) = \frac{\operatorname{cd}(z)\operatorname{cd}(w) - \operatorname{sn}(z)\operatorname{sn}(w)}{1 - k^2\operatorname{sn}(z)\operatorname{sn}(w)\operatorname{cd}(z)\operatorname{cd}(w)},$$

using Glaisher’s notation for the quotient $\operatorname{cd}(z) = \operatorname{cn}(z)/\operatorname{dn}(z) = \operatorname{sn}(z + K)$, with the complete elliptic integral $K = K(k)$ being a quarter period of the Jacobi sine, which yields (2) when we parametrize the points on E_d by

$$(x, y) = (\operatorname{sn}(z), \operatorname{cd}(z)) = (\operatorname{sn}(z), \operatorname{sn}(z + K)) \tag{3}$$

and identify $d = k^2$.

It was shown in [1] that, compared with the Weierstrass representation and its variants, the Edwards addition law gives more efficient formulae for computing an addition step $(P_1, P_2) \mapsto P_1 + P_2$ or a doubling step $P_1 \mapsto 2P_1$, both of which are required to obtain the scalar multiple sP in subexponential time $O(\log s)$ via an addition chain. The implementation EECM-MPFQ introduced in [2] gains even greater efficiency by using twisted Edwards curves, with an extra parameter a in front of the term x^2 on the left-hand side of (1), and further optimizing the ECM in other ways, including the use of projective coordinates in \mathbb{P}^2 , extended Edwards coordinates in \mathbb{P}^3 , and choosing curves with large torsion.

In this chapter, we explore implementations of the ECM using other models of elliptic curves, which arise in the context of QRT maps, an 18-parameter family of birational maps of the plane introduced by Quispel, Roberts, and Thompson [30] to unify diverse examples of maps and functional relations appearing in dynamical systems, statistical mechanics, and soliton theory. A QRT map is one of the simplest examples of a discrete integrable system, being a discrete avatar of a Hamiltonian system with one degree of freedom, with an invariant function (conserved quantity) and an invariant measure (symplectic form) [5].

Each orbit of a QRT map corresponds to a sequence of points $P_0 + nP$ on a curve of genus one, and in the special case $P_0 = O$, the orbit consists of the scalar multiples nP , being closely related to an elliptic divisibility sequence (EDS) [34]. Thus, we can implement the ECM by iterating a QRT map with a special choice of initial data and performing all the arithmetic in $\mathbb{Z}/N\mathbb{Z}$.

A terse overview of QRT maps is provided in the next section; see [5, 20, 21, 33] for further details. Section 3 briefly introduces Somos sequences and related EDS, showing how three particular examples of QRT maps arise in this context, namely the Somos-4 QRT map, the Somos-5 QRT map, and the Lyness map. Each of the subsequent Sects. 4–6 is devoted to one of these three types of QRT map, including the doubling map that sends any point $P_1 \mapsto 2P_1$ and a corresponding version of the ECM. In Sect. 7, we analyse the complexity of scalar multiplication, concentrating on the Lyness case in projective coordinates, and the final section contains some conclusions.

2 A Brief Review of QRT Maps

A QRT map can be constructed from a biquadratic curve of the general form

$$F(x, y) := \sum_{i,j=0}^2 a_{ij} x^i y^j = 0. \quad (4)$$

For generic coefficients a_{ij} , this is a smooth affine curve, and with the inclusion of additional points at infinity, it lifts to a smooth curve in $\mathbb{P}^1 \times \mathbb{P}^1$, by introducing homogeneous coordinates $((X : W), (Y : Z))$ and setting $x = X/W$, $y = Y/Z$ to obtain a homogeneous equation of bidegree $(2, 2)$, that is,

$$\hat{F}(X, W, Y, Z) = W^2 Z^2 F(X/W, Y/Z) = 0;$$

this curve is a double cover of \mathbb{P}^1 with four branch points and so has genus one by Riemann-Hurwitz. A biquadratic curve admits two simple involutions, namely the horizontal/vertical switches given by

$$\iota_h : (x, y) \mapsto (x^\dagger, y), \quad \iota_v : (x, y) \mapsto (x, y^\dagger),$$

where x^\dagger is the conjugate root of (4), viewed as a quadratic in x , and similarly for y^\dagger ; the Vieta formulae for the sum/product of the roots of a quadratic allow explicit birational expressions to be given for these two involutions. On a given biquadratic curve, the QRT map is defined to be the composition of the two switches,

$$\varphi_{\text{QRT}} = \iota_v \circ \iota_h,$$

which acts as a translation in the group law of the curve, $\varphi_{\text{QRT}} : P_0 \mapsto P_0 + P$, where the shift P is independent of the choice of initial point P_0 on the curve.

So far, the map φ_{QRT} is restricted to a single curve, but to define a map on the plane, one should allow each coefficient $a_{ij} = a_{ij}(\lambda)$ to be a linear function of a parameter λ , so that (4) becomes a biquadratic pencil,

$$E_\lambda : \quad F(x, y) \equiv F_1(x, y) + \lambda F_2(x, y) = 0. \quad (5)$$

The map $(x, y) \mapsto \lambda = -F_1(x, y)/F_2(x, y)$, obtained by solving (5) for λ , defines an elliptic fibration of the plane over \mathbb{P}^1 (except at finitely many base points where $F_1 = F_2 = 0$). Each value of λ corresponds to a unique curve in the pencil, where the map φ_{QRT} is defined, and on each such curve, a suitable combination of Vieta formulae yields a birational expression, which is independent of λ , so defines a birational map on the (x, y) plane, also denoted φ_{QRT} . By construction, the function $-F_1/F_2$ is constant on each orbit and so is a conserved quantity for the map φ_{QRT} in the plane.

Henceforth, we restrict to the symmetric case $F(x, y) = F(y, x)$, so that each curve in the pencil also admits the involution

$$\iota : (x, y) \mapsto (y, x),$$

making the horizontal/vertical switches conjugate to one another; thus, φ_{QRT} is a perfect square: $\iota_v = \iota \circ \iota_h \circ \iota$, hence $\varphi_{\text{QRT}} = (\iota \circ \iota_h)^2 = \varphi \circ \varphi$, where the “square root” of φ_{QRT} is the symmetric QRT map

$$\varphi = \iota \circ \iota_h.$$

As a simple example, note that the Edwards curve (1) is a symmetric biquadratic, and we can identify $d = \lambda$ as the parameter of the pencil. Then, the Vieta formula for the sum of the roots gives an expression that is independent of this parameter, and the symmetric QRT map $\varphi = \varphi_{\text{Edwards}}$ associated with this pencil has the very simple form

$$\varphi_{\text{Edwards}} : (x, y) \mapsto (y, -x),$$

which is periodic with period four, i.e. $(\varphi_{\text{Edwards}})^4 = \text{id}$. This is another manifestation of the well-known fact that Edwards curves have 4-torsion or of the fact that the complete elliptic integral K in (3) is a quarter period of the Jacobi sine.

A generic symmetric QRT map is far from being so simple: starting from an initial point P_0 in the plane, each orbit is a sequence of points $P_n = P_0 + nP$ on a particular curve E_λ , and in general (at least, over an infinite field), the shift P need not be a torsion point. Even over a finite field \mathbb{F}_p , where every point is torsion, the order of P typically varies with the choice of curve in the pencil, i.e. with the value of λ .

In the cases of interest for the rest of the chapter, the symmetric QRT map φ can be written in multiplicative form, so that the sequence of points P_n has coordinates $(x, y) = (u_n, u_{n+1})$, where u_n satisfies a recurrence of second order,

$$u_{n+2} u_n = R(u_{n+1}), \tag{6}$$

for a certain rational function R of degree at most two, with coefficients that are independent of λ (cf. Proposition 2.5 in [15], or [20, 21], for more details).

3 Somos and Elliptic Divisibility Sequences

A Somos- k sequence satisfies a quadratic recurrence of the form

$$\tau_{n+k}\tau_n = \sum_{j=1}^{\lfloor k/2 \rfloor} \alpha_j \tau_{n+k-j}\tau_{n+j}, \tag{7}$$

where (to avoid elementary cases) it is assumed that $k \geq 4$ with at least two parameters $\alpha_j \neq 0$. It was a surprising empirical observation of Somos [31] that such rational recurrences can sometimes generate integer sequences, which was proved by Malouf [26] for the Somos-4 recurrence

$$\tau_{n+4}\tau_n = \alpha \tau_{n+3}\tau_{n+1} + \beta (\tau_{n+2})^2, \tag{8}$$

in the particular case that the coefficients are $\alpha = \beta = 1$ and the initial values are $\tau_0 = \tau_1 = \tau_2 = \tau_3 = 1$. A broader understanding came from the further observation that the recurrence (8) has the Laurent property [10], that is, $\tau_n \in \mathbb{Z}[\alpha, \beta, \tau_0^{\pm 1}, \tau_1^{\pm 1}, \tau_2^{\pm 1}, \tau_3^{\pm 1}] \forall n \in \mathbb{Z}$. Somos sequences arise from mutations in cluster algebras [9] or LP algebras [23] and as reductions of the bilinear discrete KP/BKP equations, being associated with translations on Jacobian/Prym varieties for the spectral curve of a corresponding Lax matrix [7, 17].

The three simplest non-trivial examples of Somos recurrences, with two terms on the right-hand side, are the Somos-4 recurrence (8), the Somos-5 recurrence

$$\tau_{n+5}\tau_n = \tilde{\alpha} \tau_{n+4}\tau_{n+1} + \tilde{\beta} \tau_{n+3}\tau_{n+2}, \tag{9}$$

and the special Somos-7 recurrence

$$\tau_{n+7}\tau_n = a \tau_{n+6}\tau_{n+1} + b \tau_{n+4}\tau_{n+3}. \tag{10}$$

All three of them can be reduced to two-dimensional maps of QRT type, and hence their orbits correspond to sequences of points $P_0 + nP$ on curves of genus one. (In contrast, generic Somos-6 sequences and Somos-7 sequences are associated with points on Jacobians of genus 2 curves [7].)

To see the connection with QRT maps, in (8), one should substitute

$$u_n = \frac{\tau_{n-1}\tau_{n+1}}{\tau_n^2} \implies u_{n+2}u_n = \frac{\alpha u_{n+1} + \beta}{(u_{n+1})^2}, \tag{11}$$

yielding a second-order recurrence that can be reinterpreted as the map

$$(u_n, u_{n+1}) \mapsto (u_{n+1}, u_{n+2})$$

in the plane, and it turns out to be a symmetric QRT map; for the associated biquadratic pencil and other details, see Sect. 4. Similarly, for the Somos-5 recurrence (9), one can make the substitution

$$u_n = \frac{\tau_{n-2}\tau_{n+1}}{\tau_{n-1}\tau_n} \implies u_{n+2}u_n = \frac{\tilde{\alpha}u_{n+1} + \tilde{\beta}}{u_{n+1}}, \tag{12}$$

where the latter recurrence for u_n is equivalent to the QRT map described in Sect. 5. Finally, for the special Somos-7 recurrence (10), one should substitute

$$u_n = \frac{\tau_{n-3}\tau_{n+2}}{\tau_{n-1}\tau_n} \implies u_{n+2}u_n = a u_{n+1} + b, \tag{13}$$

reducing the order from seven to two. The recurrence for u_n in (13) is known in the literature as the Lyness map, after the particular periodic case $b = a^2$ found in [25]; for details, see Sect. 6. The first two of these substitutions were derived in an ad hoc way in [14] and [15], but they all have a very natural interpretation in the theory of cluster algebras [8], which implies that these are the only Somos- k recurrences that can be reduced to two-dimensional maps.

Morgan Ward’s elliptic divisibility sequences (EDSs) [34] are sequences of integers τ_n with $\tau_0 = 0, \tau_1 = 1, \tau_2, \tau_3, \tau_4 \in \mathbb{Z}$, and $\tau_2|\tau_4$, subject to the relations

$$\tau_{n+m}\tau_{n-m} = (\tau_m)^2\tau_{n+1}\tau_{n-1} - \tau_{m+1}\tau_{m-1}(\tau_n)^2, \tag{14}$$

$$\tau_2\tau_{n+m+1}\tau_{n-m} = \tau_{m+1}\tau_m\tau_{n+2}\tau_{n-1} - \tau_{m+2}\tau_{m-1}\tau_{n+1}\tau_n \tag{15}$$

for all $m, n \in \mathbb{Z}$. An EDS corresponds to a sequence of points nP on an elliptic curve over \mathbb{Q} . The relation (14) for $m = 2$ is a special case of the Somos-4 recurrence (8), with $\alpha = (\tau_2)^2$ and $\beta = -\tau_3$; similarly, (15) with $m = 2$ gives a special case of (9), and a linear combination of this relation for $m = 2$ and $m = 3$ yields (10) with the coefficients/initial values related in a particular way. The fact that the same EDS satisfies these higher Somos relations [29] provides one way to derive the isomorphisms between the associated biquadratic curves and a Weierstrass cubic in Theorem 1 below, which can also be deduced from results in [18].

4 Somos-4 QRT Map

Here, we give further details of the QRT map defined by (11) and the associated family of curves.

QRT map : $\varphi : (x, y) \mapsto \left(y, (\alpha y + \beta)/(xy^2) \right).$ (16)

Pencil of curves : $x^2y^2 + \alpha(x + y) + \beta - Jxy = 0.$ (17)

Elliptic involution : $\iota_E : (x, y) \mapsto \left(x, (\alpha x + \beta)/(x^2y) \right).$ (18)

Identity element and shift : $O = (\infty, 0), \quad P = (0, -\beta/\alpha).$ (19)

Doubling map : $\psi : (x, y) \mapsto$

$$\left(\frac{\alpha(x - y)y(\alpha x + \beta - x^3y)}{(\alpha x + \beta - x^2y^2)^2}, -\frac{(\alpha x + \beta - x^2y^2)(\alpha y + \beta - x^2y^2)}{\alpha xy(x - y)^2} \right). \quad (20)$$

The map (16) preserves the symplectic form $\omega = (xy)^{-1}dx \wedge dy$, that is, $\varphi^*(\omega) = \omega$, and the doubling map ψ gives $\psi^*(\omega) = 2\omega$; the same is true for the Somos-5/Lyness maps. Each orbit of φ lies on a fixed biquadratic curve of the form (17), with $\lambda = -J$ being the parameter of the pencil (5); equivalently, solving (17) for $J = J(x, y)$ gives a conserved quantity for the map. On any curve (17), the elliptic involution (18) sends any point $P \mapsto -P$. A special sequence of points (u_n, u_{n+1}) on the curve is generated by iterating (16) with a suitable starting point, corresponding to the scalar multiples nP of a particular point P (the shift). To have both coordinates finite and non-zero, one should start with

$$2P = (-\beta/\alpha, -\alpha(\alpha^2 + \beta J)/\beta^2) = (u_2, u_3). \quad (21)$$

However, in order to calculate a particular scalar multiple sP in time $O(\log s)$, rather than $O(s)$, one must employ the doubling map on the curve, using some variant of the “double-and-add” method (an addition chain).

We can now present a version of the ECM based on the QRT map (16).

Algorithm 1 ECM with Somos-4 QRT *To factorize N , pick $\alpha, \beta, J \in \mathbb{Z}/N\mathbb{Z}$ at random and some integer $s > 2$. Then, starting from the point $2P = (u_2, u_3)$ on the curve (17), given by (21), use the QRT map (16) to perform addition steps and (20) to perform doubling steps, working in $\mathbb{Z}/N\mathbb{Z}$, to compute $sP = (u_s, u_{s+1})$. Stop if, for some denominator D , $g = \gcd(D, N) > 1$ appears at any stage; when $g < N$, the algorithm has been successful, but if $g = N$ or no forbidden divisions appear, then restart with new α, β, J , and/or a larger value of s .*

Example 1 Given $N = 1,950,153,409$, we pick $\alpha = \beta = 1$ and $J = 4$ to find $(u_2, u_3) = (-1, -5)$, take $s = 12$, and compute the sequence $(u_n \bmod N)$, that is,

$$\infty, 0, -1, -5, 1482116591, 121884579, 452175879, 1062558798, 154165861, 1566968710, 1329544730, 56956778,$$

where the last term is u_{11} ; then, $g = \gcd(u_{11}, N) = 16,433$, so the algorithm terminates. Of course, not all the above terms are necessary, since by writing $12 = 2^2 \times (2 + 1)$, it is more efficient to compute the addition chain $2P \mapsto 3P \mapsto 6P \mapsto 12P$ using (16) and (20) as

$$(u_2, u_3) \xrightarrow{\varphi} (u_3, u_4) \xrightarrow{\psi} (u_6, u_7) \xrightarrow{\psi} ???$$

and then observe that the denominator $\alpha x + \beta - x^2y^2$ in (20) has common factor $g > 1$ with N when $(x, y) = (u_6, u_7)$.

5 Somos-5 QRT Map

Here, we describe the features of the QRT map corresponding to recurrence (12).

$$\text{QRT map : } \quad \varphi : (x, y) \mapsto \left(y, (\tilde{\alpha}y + \tilde{\beta})/(xy) \right). \quad (22)$$

$$\text{Pencil of curves : } \quad xy(x+y) + \tilde{\alpha}(x+y) + \tilde{\beta} - \tilde{J}xy = 0. \quad (23)$$

$$\text{Elliptic involution : } \quad \iota_E : (x, y) \mapsto (y, x).$$

$$\text{Identity element and shift : } \quad O = (\infty, \infty), \quad P = (\infty, 0).$$

$$\text{Doubling map : } \quad \psi : (x, y) \mapsto \left(\frac{(x^2y - \tilde{\alpha}x - \tilde{\beta})(x^2y - \tilde{\alpha}y - \tilde{\beta})}{x(x-y)(xy^2 - \tilde{\alpha}x - \tilde{\beta})}, \frac{(xy^2 - \tilde{\alpha}x - \tilde{\beta})(xy^2 - \tilde{\alpha}y - \tilde{\beta})}{y(y-x)(x^2y - \tilde{\alpha}y - \tilde{\beta})} \right). \quad (24)$$

The double of the translation point (shift) is $2P = (0, -\tilde{\beta}/\tilde{\alpha}) = (u_2, u_3)$, so to obtain the sequence of multiples nP , one must start with

$$3P = (-\tilde{\beta}/\tilde{\alpha}, \tilde{J} + \tilde{\alpha}^2/\tilde{\beta} + \tilde{\beta}/\tilde{\alpha}) = (u_3, u_4). \quad (25)$$

We can paraphrase Algorithm 1 to get another version of the ECM.

Algorithm 2 ECM with Somos-5 QRT *To factorize N , pick $\tilde{\alpha}, \tilde{\beta}, \tilde{J} \in \mathbb{Z}/N\mathbb{Z}$ at random and some integer $s > 3$. Then, starting from $3P = (u_3, u_4)$ on the curve (23), given by (25), use (22) to perform addition steps and (24) to perform doubling steps, working in $\mathbb{Z}/N\mathbb{Z}$, to compute $sP = (u_s, u_{s+1})$. Stop if, for some denominator D , $g = \gcd(D, N)$ with $1 < g < N$ appears at any stage.*

6 Lyness Map

The real and complex dynamics of the recurrence (13), known as the Lyness map, has been studied by many authors, with a very detailed account in [5].

$$\text{QRT map : } \quad \varphi : (x, y) \mapsto \left(y, \frac{ay+b}{x} \right). \quad (26)$$

Pencil of curves :

$$xy(x+y) + a(x+y)^2 + (a^2+b)(x+y) + ab - Kxy = 0. \quad (27)$$

$$\text{Elliptic involution : } \quad \iota_E : (x, y) \mapsto (y, x).$$

$$\text{Identity element and shift : } \quad O = (\infty, \infty), \quad P = (\infty, -a). \quad (28)$$

Doubling map : $\psi : (x, y) \mapsto (R(x, y), R(y, x)),$

$$R(x, y) = \frac{(xy - ay - b)(x^2y - a^2x - by - ab)}{x(x - y)(y^2 - ax - b)}. \tag{29}$$

Doubling and tripling P give $2P = (-a, 0), 3P = (0, -b/a),$ so to obtain the multiples $nP = (u_n, u_{n+1})$ by iteration of (26) and (29), one should begin with

$$4P = \left(-\frac{b}{a}, -a - \frac{b(Ka + b)}{a(a^2 - b)} \right) = (u_4, u_5). \tag{30}$$

Henceforth, it will be assumed that $b \neq a^2,$ since otherwise all orbits of (13) are periodic with period five, meaning that P is a 5-torsion point on every curve in the pencil. This special case is the famous Lyness 5-cycle [25], related to the associahedron K_4 via the A_2 cluster algebra and to the Abel pentagon identity for the dilogarithm [27], among many other things.

The above formulae (and those for Somos-4/5) can all be obtained via the birational equivalence of curves described in the following theorem (cf. [18]).

Theorem 1 *Given a fixed choice of rational point $P = (v, \xi) \in \mathbb{Q}^2$ on a Weierstrass cubic*

$$E(\mathbb{Q}) : (y')^2 = (x')^3 + Ax' + B$$

over $\mathbb{Q},$ a point (x, y) on a Lyness curve (27) is given in terms of $(x', y') \in E(\mathbb{Q})$ by

$$x = -\frac{\beta(\alpha u + \beta)}{uv} - a, \quad y = -\beta uv - a,$$

where

$$(u, v) = \left(v - x', \frac{4\xi y' + Ju - \alpha}{2u^2} \right)$$

are the coordinates of a point on the Somos-4 curve (17), and the parameters are related by

$$a = -\alpha^2 - \beta J, \quad b = 2\alpha^2 + a\beta J - \beta^3, \quad K = -2a - \beta J, \tag{31}$$

with

$$\alpha = 4\xi^2, \quad J = 6v^2 + 2A, \quad \beta = \frac{1}{4}J^2 - 12v\xi^2.$$

Also,

$$\left(-\frac{x+a}{\beta}, -\frac{y+a}{\beta}\right)$$

is a point on the Somos-5 curve (23) with parameters

$$\tilde{\alpha} = -\beta, \quad \tilde{\beta} = \alpha^2 + \beta J, \quad \tilde{J} = J.$$

Conversely, given $a, b, K \in \mathbb{Q}$, a point (x, y) on (27) corresponds to $(\bar{x}, \bar{y}) \in \bar{E}(\mathbb{Q})$, a twist of $E(\mathbb{Q})$ with coefficients $\bar{A} = \alpha^2 \beta^4 A$ and $\bar{B} = \alpha^3 \beta^6 B$, and P in (28) corresponds to the point $(\bar{v}, \bar{\xi}) = (\frac{1}{12}(\beta J)^2 - \frac{1}{3}\beta^3, \frac{1}{2}\alpha^2 \beta^3) \in \bar{E}(\mathbb{Q})$.

Algorithm 3 ECM with Lyness To factorize N , pick $a, b, K \in \mathbb{Z}/N\mathbb{Z}$ at random and some integer $s > 4$. Then, starting from $4P = (u_4, u_5)$ on the curve (27), given by (30), use (26) to perform addition steps and (29) to perform doubling steps, working in $\mathbb{Z}/N\mathbb{Z}$, to compute $sP = (u_s, u_{s+1})$. Stop if, for some denominator D , $g = \gcd(D, N)$ with $1 < g < N$ appears at any stage.

7 Complexity of Scalar Multiplication

Of the three symmetric QRT maps above, the Lyness map (26) is the simplest, so we focus on that for our analysis. Before proceeding, we can make the simplification $a \rightarrow 1$ without loss of generality, since over \mathbb{Q} we can always rescale $(x, y) \rightarrow (ax, ay)$ and redefine b and K . To have an efficient version of Algorithm 3, it is necessary to work in projective coordinates, to avoid costly modular inversions; then, only a single gcd needs to be calculated at the end. For cubic curves, it is most common to work in the projective plane \mathbb{P}^2 (or sometimes, Jacobian coordinates in the weighted projective space $\mathbb{P}(1, 2, 3)$ are used for Weierstrass cubics [11]). However, for the biquadratic cubics (27), $\mathbb{P}^1 \times \mathbb{P}^1$ is better, since doubling with (29) is of higher degree in \mathbb{P}^2 .

In terms of projective coordinates in $\mathbb{P}^1 \times \mathbb{P}^1$, the Lyness map (26) becomes

$$\left((X : W), (Y : Z)\right) \mapsto \left((Y : Z), ((aY + bZ)W : XZ)\right). \tag{32}$$

Then, taking $a \rightarrow 1$, each addition step using (32) requires $2M + 1B$, i.e. two multiplications and one multiplication by parameter b .

The affine doubling map (29) for the Lyness case lifts to the projective version

$$\left((X : W), (Y : Z)\right) \mapsto \left((A_1 B_1 : C_1 D_1), (A_2 B_2 : C_2 D_2)\right), \tag{33}$$

where

$$X^* = A_1 B_1, \quad W^* = C_1 D_1, \quad Y^* = A_2 B_2, \quad Z^* = C_2 D_2,$$

$$A_1 = A_+ + A_-, \quad A_2 = A_+ - A_-, \quad B_1 = B_+ + B_-, \quad B_2 = B_+ - B_-,$$

$$C_1 = 2XT, \quad C_2 = -2YT, \quad D_1 = ZA_2 + C_2, \quad D_2 = WA_1 + C_1,$$

with $A_- = aT$ and

$$A_+ = 2G - aS - 2H', \quad B_+ = S(G - a^2H - H') - 2aHH', \quad B_- = T(G - a^2H + H'),$$

$$S = E + F, \quad T = E - F, \quad E = XZ, \quad F = YW, \quad G = XY, \quad H = WZ, \quad H' = bH.$$

Setting $a \rightarrow 1$ once again for convenience and using the above formulae, we see that doubling can be achieved with $15\mathbf{M} + 1\mathbf{B}$. (To multiply by 2, use addition: $2X = X + X$.)

This should be compared with EECM-MPFQ [2]: using twisted Edwards curves $ax^2 + y^2 = 1 + dx^2y^2$ in \mathbb{P}^2 , the projective addition formula requires $10\mathbf{M} + 1\mathbf{S} + 1\mathbf{A} + 1\mathbf{D}$ (\mathbf{S} , \mathbf{A} , and \mathbf{D} denote squaring and multiplication by the parameters a and d , respectively), while doubling only takes $3\mathbf{M} + 4\mathbf{S} + 1\mathbf{A}$. So, the Lyness addition step (32) is much more efficient than for twisted Edwards, but doubling requires twice as many multiplications. For any addition chain, the number of doublings will be $O(\log s)$, so employing Algorithm 3 to carry out the ECM with the Lyness map in projective coordinates should require on average roughly twice as many multiplications per bit as for EECM-MPFQ.

8 Conclusions

Due to the complexity of doubling, it appears that scalar multiplication with Lyness curves is not competitive with the state of the art using twisted Edwards curves. However, in a follow-up study [19], we have shown that the projective doubling map (33) for Lyness curves can be made efficient by distributing it over four processors in parallel, dropping the effective cost to $4\mathbf{M} + 1\mathbf{B}$. On the other hand, this is still roughly twice the cost of the best known algorithm for doubling with four processors on twisted Edwards curves in the special case $a = -1$ [13].

However, by Theorem 1, any elliptic curve over \mathbb{Q} is isomorphic to a Lyness curve, while twisted Edwards curves only correspond to a subset of such curves. Thus, there may be other circumstances, whether for the ECM or for alternative cryptographic applications, where Lyness curves and QRT maps will prove to be useful. For instance, one could use families of Lyness curves with torsion subgroups that are impossible with twisted Edwards curves in EECM-MPFQ. Also, bitcoin uses the curve $y^2 = x^3 + 7$, known as secp256k1, which cannot be expressed in twisted Edwards form.

The remarkable simplicity of the addition step (32) means that it might also be suitable for pseudorandom number generation. In that context, it would be worth exploring non-autonomous versions of QRT maps mod N . For example, the recurrence

$$u_{n+2}u_n = u_{n+1} + b_nq^n, \quad b_{n+6} = b_n \quad (34)$$

is a q -difference Painlevé version of the Lyness map (13) (see [16]), and over \mathbb{Q} , the arithmetic behaviour of such equations appears to be analogous to the autonomous case [12], with polynomial growth of logarithmic heights; although for (34), the growth is cubic rather than quadratic as in the elliptic curve case. It is interesting to compare this with the case where $q = 1$ and the coefficient b_n is periodic with a period that does not divide 6, when generically (34) appears to display chaotic dynamics [3], e.g. the period 5 example $u_{n+2}u_n = u_{n+1} + b_n$, $b_{n+5} = b_n$, for which the logarithmic height along orbits in \mathbb{Q} grows exponentially with n . Working mod N , it would be worth carrying out a comparative study of the pseudorandom sequences generated by (34) to see how the behaviour for $q \neq 1$ differs from the Lyness case (13) and the effect of changing the period of b_n .

Acknowledgments This research was supported by Fellowship EP/M004333/1 from the Engineering & Physical Sciences Research Council, UK. The author thanks the School of Mathematics and Statistics, University of New South Wales, for hosting him twice during 2017–2019 as a Visiting Professorial Fellow, with funding from the Distinguished Researcher Visitor Scheme. He is also grateful to John Roberts and Wolfgang Schief for providing additional support during his stay in Sydney, where the idea behind this work originated, and to Reinout Quispel for useful discussions and hospitality during his visit to Melbourne in May 2019.

References

1. D.J. Bernstein, T. Lange, Faster addition and doubling on elliptic curves, in ed. by K. Kurosawa, *Advances in Cryptology – ASIACRYPT 2007* (Springer, Berlin, 2007), pp. 29–50. https://doi.org/10.1007/978-3-540-76900-2_3
2. D.J. Bernstein, P. Birkner, T. Lange, C. Peters, ECM using Edwards curves. *Math. Comput.* **82**, 1139–1179 (2013). <https://doi.org/10.1090/S0025-5718-2012-02633-0>
3. A. Cima, A. Gasull, V. Mañosa, Integrability and non-integrability of periodic non-autonomous Lyness recurrences. *Dyn. Syst.* **28**, 518–538 (2013). <https://doi.org/10.1080/14689367.2013.821103>
4. R. Crandall, C. Pomerance, *Prime Numbers - A Computational Perspective*, 2nd edn (Springer, New York, 2005)
5. J.J. Duistermaat, *Discrete Integrable Systems: QRT Maps and Elliptic Surfaces* (Springer, New York, 2010)
6. H.M. Edwards, A normal form for elliptic curves. *Bull. Amer. Math. Soc.* **44**, 393–422 (2007). <https://doi.org/10.1090/S0273-0979-07-01153-6>
7. Y.N. Fedorov, A.N.W. Hone, Sigma-function solution to the general Somos-6 recurrence via hyperelliptic Prym varieties. *J. Integrable Syst.* **1**, xyw012 (2016). <https://doi.org/10.1093/integr/xyw012>

8. A.P. Fordy, A.N.W. Hone, Discrete integrable systems and Poisson algebras from cluster maps. *Commun. Math. Phys.* **325**, 527–584 (2014). <https://doi.org/10.1007/s00220-013-1867-y>
9. A.P. Fordy, R.J. Marsh, Cluster mutation-periodic quivers and associated Laurent sequences. *J. Algebraic Combin.* **34**, 19–66 (2011). <https://doi.org/10.1007/s10801-010-0262-4>
10. D. Gale, The strange and surprising saga of the Somos sequences. *Math. Intell.* **13**(1), 40–42 (1991); Somos sequence update, *Math. Intell.* **13**(4), 49–50 (1991). Reprinted in D. Gale, *Tracking the Automatic Ant* (Springer, New York, 1998)
11. R.R. Goundar, M. Joye, A. Miyaji, Co-Z addition formulae and binary ladders on elliptic curves, in ed. by S. Mangard, F.-X. Standaert, *Cryptographic Hardware and Embedded Systems, CHES 2010*. Lecture Notes in Computer Science, vol. 6225. (Springer, Berlin, 2010), pp. 65–79. https://doi.org/10.1007/978-3-642-15031-9_5
12. R.G. Halburd, Diophantine integrability. *J. Phys. A Math. Gen.* **38**, L1–L7 (2005). <https://doi.org/10.1088/0305-4470/38/16/L01>
13. H. Huseyin, K.K.-H. Wong, G. Carter, E. Dawson, Twisted Edwards curves revisited, in ed. by J. Pieprzyk, *Advances in Cryptology - ASIACRYPT 2008*. Lecture Notes in Computer Science, vol. 5350 (2008), pp. 326–343. https://doi.org/10.1007/978-3-540-89255-7_20
14. A.N.W. Hone, Elliptic curves and quadratic recurrence sequences. *Bull. Lond. Math. Soc.* **37**, 161–171 (2005). <https://doi.org/10.1112/S0024609304004163>. Corrigendum. *Bull. Lond. Math. Soc.* **38**, 741–742 (2006). <https://doi.org/10.1112/S0024609306018844>
15. A.N.W. Hone, Sigma function solution of the initial value problem for Somos 5 sequences. *Trans. Amer. Math. Soc.* **359**, 5019–5034 (2007). <https://doi.org/10.1090/S0002-9947-07-04215-8>
16. A.N.W. Hone, R. Inoue, Discrete Painlevé equations from Y-systems. *J. Phys. A: Math. Theor.* **47**, 474007 (2014). <https://doi.org/10.1088/1751-8113/47/47/474007>
17. A.N.W. Hone, T.E. Kouloukas, C. Ward, On reductions of the Hirota-Miwa equation. *SIGMA* **13**, 057 (2017). <https://doi.org/10.3842/SIGMA.2017.057>
18. A.N.W. Hone, C.S. Swart, Integrality and the Laurent phenomenon for Somos 4 and Somos 5 sequences. *Math. Proc. Camb. Phil. Soc.* **145**, 65–85 (2008). <https://doi.org/10.1017/S030500410800114X>
19. A.N.W. Hone, Efficient ECM factorization in parallel with the Lyness map (2020). arXiv:2002.03811
20. A. Iatrou, J.A.G. Roberts, Integrable mappings of the plane preserving biquadratic invariant curves. *J. Phys. A: Math. Gen.* **34**, 6617–6636 (2001). <https://doi.org/10.1088/0305-4470/34/34/308>
21. A. Iatrou, J.A.G. Roberts, Integrable mappings of the plane preserving biquadratic invariant curves II. *Nonlinearity* **15**, 459–489 (2002). <https://doi.org/10.1088/0951-7715/15/2/313>
22. N. Koblitz, *Algebraic Aspects of Cryptography* (Springer, Berlin, 1998)
23. T. Lam, P. Pylyavskyy, Laurent phenomenon algebras. *Cam. J. Math.* **4**, 121–162 (2012). <https://doi.org/10.4310/CJM.2016.v4.n1.a2>
24. H.W. Lenstra, Jr., Factoring integers with elliptic curves. *Ann. Math.* **126**, 649–673 (1987). <https://doi.org/10.2307/1971363>
25. R.C. Lyness, Cycles. *Math. Gaz.* **26**, 62 (1942)
26. J.L. Malouf, An integer sequence from a rational recursion. *Discrete Math.* **110**, 257–261 (1992). [https://doi.org/10.1016/0012-365X\(92\)90714-Q](https://doi.org/10.1016/0012-365X(92)90714-Q)
27. T. Nakanishi, Periodicities in cluster algebras and dilogarithm identities, in ed. by A. Skowronski, K. Yamagata, *Representations of Algebras and Related Topics, EMS Series of Congress Reports* (European Mathematical Society, Zurich, 2011), pp. 407–444
28. F.W.J. Olver, A.B. Olde Daalhuis, D.W. Lozier, B.I. Schneider, R.F. Boisvert, C.W. Clark, B.R. Miller, B.V. Saunders, H.S. Cohl, M.A. McClain, (Eds.), *NIST Digital Library of Mathematical Functions*. <http://dlmf.nist.gov/>. Release 1.0.25 of 2019-12-15
29. A.J. van der Poorten, C.S. Swart, Recurrence relations for elliptic sequences: every Somos 4 is a Somos k . *Bull. London Math. Soc.* **38**, 546–554 (2006). <https://doi.org/10.1112/S0024609306018534>

30. G.R.W. Quispel, J.A.G. Roberts, C.J. Thompson, Integrable mappings and soliton equations. *Phys. Lett. A* **126**, 419–421 (1988)
31. M. Somos, Problem 1470. *Crux Math.* **15**, 208 (1989)
32. D.R. Stinson, *Cryptography Theory and Practice*, 3rd edn (Chapman & Hall/CRC, Boca Raton, 2006)
33. T. Tsuda, Integrable mappings via rational elliptic surfaces. *J. Phys. A: Math. Gen.* **37**, 2721–2730 (2004). <https://doi.org/10.1088/0305-4470/37/7/014>
34. M. Ward, Memoir on elliptic divisibility sequences. *Amer. J. Math.* **70**, 31–74 (1948). <https://doi.org/10.2307/2371930>
35. E.T. Whittaker, G.N. Watson, *A Course of Modern Analysis*, 4th edn. (Cambridge University Press, Cambridge, 1927)
36. S.Y. Yan, *Primality Testing and Integer Factorization in Public-Key Cryptography* (Kluwer Academic Publishers, Boston, 2004)

What Have Google’s Random Quantum Circuit Simulation Experiments Demonstrated About Quantum Supremacy?



Jack K. Horner and John F. Symons

1 What Is Quantum Supremacy?

Quantum computing is of high interest because it promises to perform at least some kinds of computations much faster than classical computers. Quantum computers can execute some tasks “faster” than classical computers¹ only if those tasks can be executed “concurrently”.² For example, the search for prime numbers can be executed concurrently (see, e.g., [5]). In contrast, solving some computational fluid dynamics systems requires time-ordering of computational steps (see, e.g., [16]) in the sense that there is no known computational method that would allow us to avoid that ordering.

For the purposes of this paper, we need not comprehensively characterize classical or quantum computing; we assume, however, at least the following difference between quantum and classical computers ([19], 13):

(CQD) A *quantum computer* performs a computation in a way that is describable only by implying superposition of states as defined by the quantum theory [3, 18]. A *classical*

¹For the purpose of this paper, a “classical” computer is a computer to which a finite Turing machine ([6], 44) is homomorphic.

²Informally speaking, a calculation schema can be executed concurrently if more than one instance of the schema can be executed “at the same time” on a computing system. (For a more rigorous definition, see [11], esp. Chaps. 2–4).

J. K. Horner (✉)
Independent Researcher, Lawrence, KS, USA
e-mail: jhorner@cybermesa.com

J. F. Symons
Department of Philosophy, University of Kansas, Lawrence, KS, USA

computer in contrast performs a computation that is describable in a way that does not imply superposition of states.

Informally, *quantum supremacy* is the claim (see, e.g., [4, 10]) that

(IQS) A quantum computer can perform some set of computational tasks faster than a classical computer can.

There are many ways of interpreting IQS, depending on how we characterize “task,” “faster,” and “physical computation.” The declared objective of Arute et al. [1] (p. 505) is to demonstrate what we call *Google quantum supremacy*:

(GQS) At least one computational task can be executed exponentially faster on a quantum computer than on a classical computer.

2 Overview of Arute et al. [1, 2]

The computational task (“benchmark”) used by the Google Quantum Team to demonstrate GQS is the sampling of the output of simulated pseudo-random quantum circuits. These simulated circuits are designed to entangle a set of quantum bits (qubits) by repeated application of single-qubit and two-qubit logical operations. The output of these circuits can be represented as a set of bitstrings (e.g., {000101, 11001, ...}). Because of the entanglement induced by the operations on the qubits, some bitstrings are more much likely to occur than others ([1], 506).

Sampling the output of simulated random quantum circuits is a good benchmark for evaluating GQS because random circuits possess no structure other than the total number of qubits and the operations performed on individual, and pairs, of qubits. This approach helps to maximize the generality of the results, in the best case allowing the benchmark to confirm that the problem is computationally hard (e.g., cannot be solved in non-polynomial time).

The experiments of Arute et al. [1, 2] use a 53-qubit quantum computer – which they call the “Sycamore” processor – that contains fast, high-fidelity gates that can be executed simultaneously across a two-dimensional qubit array whose qubits are connected by adjustable (“configurable”/“programmable”) couplers (see Fig. 1).

We will call a processor that has the kind of connection scheme shown in Fig. 1 a “Sycamore architecture processor.”

The Google Quantum Team evaluated the correctness of Sycamore’s computation in two general ways: (a) by comparison with the output of a supercomputer running simulations of random quantum circuits and (b) by component-level calibration of the Sycamore system, extrapolated to all configurations of interest. Some details on each of these approaches follows.

- (a) The bitstring output of Sycamore’s simulation of sampling of pseudo-random entangled quantum circuits was compared with the output of a classical supercomputer (the JUQCS-E and JUQCS-A simulators [8] on supercomputers at the Jülich Supercomputing Center) simulating the sampling of entangled

Fig. 1 Connection scheme of a 53 (working)-qubit Sycamore processor solid crosses denote qubits that work; outlined crosses denote qubits that don’t work. “Squares” represent “adjustable couplers”. (Adapted from [1], 505)



pseudo-random quantum circuits. For simulating circuits of 43 qubits or fewer, the supercomputer used in the experiments employed a Schrödinger algorithm ([2], 49–50) to compute the full quantum state.

For pseudo-random quantum circuits containing more than 43 qubits, Arute et al. [1] report there was not enough random access memory in the supercomputer used in the experiments to store the full quantum state, so the full quantum state had to be calculated by other means. On the supercomputer used, in order to simulate the state of pseudo-random quantum circuits containing 44–53 qubits, Arute et al. [1] employed a hybrid Schrödinger-Feynman algorithm [17]. The latter method breaks a circuit into two (or more) subcircuits, simulates each subcircuit with a Schrödinger method, and then “reconnects” the subcircuits using an approach in some ways analogous to the Feynman path integral [9].

The outputs obtained from the Sycamore processor and the supercomputer simulations agree well for the specific set of simulated quantum circuits analyzed by Arute et al. [1].

- (b) To further assess the results of comparison of the outputs of the Sycamore processor and the corresponding supercomputer simulations ((a), above) the Google Quantum Team calibrated the “fidelity” of the Sycamore processor for a subset of possible circuit configurations of 53 or fewer qubits. The Google Quantum Team showed that a “fidelity” model ([2], Section VIII) defined in terms of component-level error rates in a Sycamore architecture processor containing Q qubits

$$F = \prod_{g \in G_1} (1 - e_g) \prod_{g \in G_2} (1 - e_g) \prod_{q \in Q} (1 - e_q), \tag{1}$$

where there are G_1 gates of “Type 1” and G_2 gates of “Type 2,” and

e_g is the Pauli error of gate g

e_q is the state preparation and measurement error of qubit q

F is the fidelity of the processor configuration of interest

produces predictions that agree well with the fidelity Sycamore architecture systems used in the Google Quantum Team's experiments. The predictions of Eq. 1 thus at least partially corroborate the results of (a), for the cases tested by the Google Quantum Team.

If extrapolated, Eq. 1 might seem to be a fidelity/error distribution model for Sycamore architecture systems containing more than 53 qubits or for circuit configurations of 53 qubits or less that were not simulated. To this, and the more general question of whether GQS can be extrapolated to cases not yet observed, we now turn.³

3 What Do the Google Quantum Team's Experiments Show About Quantum Supremacy?

The Google Quantum Team's methods and results make a compelling case that GQS summarizes the experiments the team performed. Can GQS be extrapolated to cases *not* tested to date? Answering that question requires answering several others. Among these are the following:

1. Could an extrapolation of Eq. 1 for Sycamore architecture systems containing more than 53 qubits be grounded in statistical inference theory, based on the results reported in Arute et al. [1, 2]?
2. Does any method for comparing the performance of a quantum computer to the performance of a classical computer generalize beyond cases tested (including, but not limited to, those tested by the Google Quantum Team)?

3.1 Could an Extrapolation of Eq. 1 for Sycamore Architecture Systems Containing More Than 53 Qubits Be Grounded in Statistical Inference Theory, Based on the Results Reported in Arute et al. [1, 2]?

Let's assume that the Google Team's experiments show that Eq. 1 produces, for Sycamore architecture systems containing no more than 53 qubits, fidelity predictions that agree well enough with those of the simulators running on classical computers.

The mathematical form of Eq. 1 is identical to the mathematical form of the fidelity/error distribution of a collection of components whose error distributions are independent and identically distributed (IID; ([12], 121)). IID models are commonly

³Arute et al. [1, 2] do not explicitly claim or imply that Eq. 1 or GQS can be extrapolated for configurations other than those tested by them.

used to characterize system-level failure distributions in such as computer components, lighting equipment, fasteners, and many other mass-produced hardware items [20]. One might be tempted to infer that Eq. 1 can therefore be unproblematically extrapolated to describe the distribution of errors in Sycamore systems containing more than 53 qubits. But is this reasoning robust?

There is no question that the results of the Google Quantum Team's experiments are *consistent with* the behavior of a Sycamore architecture system containing fewer than 54 qubits whose component-level error distributions are characterized by IID. Equation 1, however, *does not imply* (in the sense of [22], 98) the component-level error distributions are IID. Why? Consider a set of component-level non-IID that in aggregate produce exactly the predictions of Eq. 1 for systems of 53 or fewer qubits. Let's call such cases Kalai configurations.⁴

The existence of Kalai configurations tells us that we have to be clear what role Eq. 1 is intended to play if it is extrapolated beyond the experimental evidence. Any such extrapolation would have to treat Eq. 1 as a distribution function in a *statistical inference* ([12], Chaps. 5–12).⁵ Statistical inference requires that its domain of application be characterizable in terms of random variables. More specifically,

(AP) Random variables are defined in terms of random experiments. In order to define a random experiment, E, we must know, *independent of experience* (i.e., a priori), all possible outcomes of that experiment. ([12], Section 1.1 and Df. 1.5.1)

As used by Arute et al. [1, 2], Eq. 1 concerns only physics-/hardware-related errors. There is another potential source of error in computing systems that is not as such derivable from the physics or hardware of the system as such. In the experiments of Arute et al. [1, 2] in particular, the process of configuring a Sycamore architecture, a prescription for “adjusting the couplers,” involves manipulating the apparatus in ways that are formally equivalent to “writing a program” or “developing software” for the processor. For economy of expression, we will call such a prescription *software*. Software in this sense can include, among other things, conditional prescriptions. For example, suppose that the couplers in a Sycamore architecture system are named A, B, C . . . and suppose also that we prescribe that the coupler between A and B must be “adjusted” *if* B is coupled to C. This kind of conditional prescription would instantiate a binary branch in the setup/software of the system.

Let S be software in the sense of the previous paragraph. Suppose S contains, on average, one binary branch per 10 instructions. Branching induces an execution path network (equivalently, an execution state-space [27]) structure on S. In order to know – as required by AP – *all* the possible outcomes of a random experiment E performed on S, we must know what S will do when each of these paths is executed. Now in general, the number of such paths in a system of M instructions, with, on average, a binary branch instruction every N instructions, is $2^{M/N}$, where $N < M$. In

⁴This is a generalization of a suggestion made by Gil Kalai (See [14]).

⁵For further detail, see Symons and Horner [25].

general, determining S 's behavior on all paths by *testing* would take $\sim 10^{13}$ lifetimes of the universe even if M is as small as 1000 and $N = 10$ (see [24–26] for further detail).

The testing-based “state-explosion” [27] problem sketched in the previous paragraph could be overcome if we could determine, without testing (i.e., a priori), the behavior of S on each path. In particular, if we had a method for automatically generating provably correct software from provably correct models/specifications, and if using this method were less onerous than exhaustively testing all paths in the resulting software, it would seem that we could dodge the state-explosion problem faced by characterization through testing.

There is such a method, generically called *model checking*, and it has contributed to some impressive practical results [7]. Notwithstanding those achievements, however, almost if not all software involving model checking to date has been produced by using software (development) systems/environments that themselves were not produced through model checking. Those software development environments typically include editors, compilers, linkers, operating systems, etc., and each of these contains thousands to millions of instructions. To the extent that these environments were not developed through model checking, we cannot infer that the software produced by using those environments is fully characterized (see [25]).⁶

We conclude, therefore, that Requirement AP cannot be satisfied by purely empirical methods. This implies, in particular, that Eq. 1 and GQS cannot, on the basis of experiments like those of Arute et al. [1, 2] per se, be extrapolated by statistical inference to cases that have not been observed.

3.2 *Does Any Method for Comparing the Performance of a Quantum Computer to the Performance of a Classical Computer Generalize Beyond the Cases Tested?*

In order to answer this question, we need to be clear about what it means to compare the performance of two physical computing systems. We posit the following condition of adequacy for any such comparison:

(CCA) Let A and B be physical computing systems. Let $C(A)$ be a computation performed on A and $C(B)$ be a computation performed on B . Let $T(A)$ be the physical theory required to completely describe the trajectory of A while performing $C(A)$. Let $T(B)$ be the physical theory required to completely describe the trajectory of B while performing $C(B)$. Then $C(A)$ and $C(B)$ can be compared only if $T(A)$ and $T(B)$ are consistent with each other.

The comparison of the performances of quantum and classical computers is not like comparing the performance of two classical computers that have the

⁶The configuration prescriptions used in the experiments of Arute et al. [1, 2] in particular were not produced in a software development environment all of which was produced by model checking.

same architecture.⁷ Why? By CQD, quantum computation requires that states can be superposed; classical computation, by definition, denies that state(s) can be superposed. Thus, by CCA, classical and quantum computations are simply not comparable.

Given the Copenhagen interpretation (CI) of quantum mechanics (see, e.g., [18], Vol. I, Chap. 4, Sections 16 and 17), furthermore, even determining the state of a quantum computer is inherently different from determining the state of a classical computer: in the CI, quantum state measurement is relativized to specific measurement configurations but classical state measurement is not. Thus, given the CI, by CCA, classical and quantum computations are not comparable.

Given these arguments, what can be the claim that the performances of a classical, and a quantum computer, at least in one case, are comparable (as, e.g., Arute et al. [1, 2] do) mean? Just this: any such a claim is only comparing the speeds at which a given classical and a given quantum computer produces given classically described outputs, given classically described inputs. That kind of claim, however, treats a computer as a (classical) “black box.” By definition, black box comparisons compare only outputs, for given inputs [21]. A black box comparison is defined only for particular pairs of outputs because, by definition, there *can be no theory that allows us to extrapolate* from the particular observations we have made to anything else.⁸

4 Conclusions

Arute et al. [1, 2] provide strong empirical evidence of GQS for the test cases considered. Given CQD in Sect. 1, Requirement AP in Sect. 3.1, however, quantum/classical computing comparisons can show at most *only* that GQS holds for the set of specific quantum/classical computer pairs that have already been observed. Given the problems of comparing classical and quantum computers discussed in Sect. 3.2, furthermore, it is not possible to compare classical and quantum computing except in a “black box” sense, and such comparisons cannot be extrapolated beyond the results of specific experiments. These results show that generalizing the claim of quantum supremacy beyond what has been observed in particular cases will continue to be a difficult problem.

Acknowledgments We would like to thank the referees for this paper for their helpful criticism. Thanks also to Francisco Pipa for his careful reading and comments. JFS was supported in part by

⁷Even defining what “same architecture” means in the case of classical computers is complicated (see [11, 15]).

⁸This problem is *not* the same as the well-known problem of induction (see, e.g., [13], Book I, Part III; [23], Section I), i.e., the problem of inferring events we haven't observed from those we have. The quantum/classical computer comparison conundrum trades only the fact that the concepts of “state” in quantum and classical contexts are incompatible.

NSA Science of Security initiative contract #H98230-18-D-0009. JKH was supported in part by a Ballantine Foundation grant.

References

1. F. Arute et al., Quantum supremacy using a programmable superconducting processor. *Nature* **574**, 505–511 (2019). <https://doi.org/10.1038/s41586-019-1666-5>. Open access. Accessed 1 January 2020
2. F. Arute, et al., Supplemental Information for Arute F et al. 2019a (2019), File `supp_info_41586_2019_1666_MOESM1_ESM.pdf`. https://static-content.springer.com/esm/art%3A10.1038%2Fs41586-019-1666-5/MediaObjects/41586_2019_1666_MOESM1_ESM.pdf. Accessed 6 Feb 2020
3. D. Bohm, *Quantum theory*. Dover reprint, 1989 (1951)
4. S. Boixo et al., Characterizing quantum supremacy in near-term devices. *Nat. Phys.* **14**, 595–600 (2018). <https://doi.org/10.1038/s41567-018-0124-x>
5. S.H. Bokhari, Multiprocessing the sieve of Eratosthenes. *IEEE Comput.* **20**(4), 50–58 (1984)
6. G.S. Boolos, J.P. Burgess, R.C. Jeffrey, *Computability and Logic*, 5th edn. (Cambridge, 2007)
7. E. M. Clarke, T. A. Henzinger, H. Veith, R. Bloem (eds.), *Handbook of Model Checking* (Springer, 2018)
8. H. De Raedt et al., Massively parallel quantum computer simulator, eleven years later. *Comput. Phys. Commun.* **237**, 57–61 (2018). <https://www.sciencedirect.com/science/article/pii/S0010465518303977?via%3DIihub>. Open access. Accessed 26 January 2020
9. C. Grosche, An introduction into the Feynman path integral. arXiv: hep-th/930209v1 (1993)
10. A. Harrow, A. Montanaro, Quantum computational supremacy. *Nature* **549**, 203–209 (2017). <https://doi.org/10.1038/nature23458>
11. J.L. Hennessy, D.A. Patterson, *Computer Architecture*, Fourth edn. (Morgan Kaufmann, 2007)
12. R.V. Hogg, J.W. McKean, A.T. Craig, *Introduction to Mathematical Statistics*, 6th edn. (Prentice Hall, 2005)
13. D. Hume, *A Treatise of Human Nature*. Ed. by L. A. Selby-Bigge. (Oxford, 1739)
14. Israeli Institute for Advanced Science, Hebrew University, Panel discussion of quantum supremacy. Part of the Mathematics of Quantum Computation Winter School (19 December 2019), <https://ias.huji.ac.il/SchoolCSE4>. You Tube. https://www.youtube.com/watch?v=H4t2G2gay7Q&feature=youtu.be&fbclid=IwAR2xkgWBPTNCJInEmk_Rak2gTm6M5yXKcV-zInW9xW2znWwJI554nXXLaug. Accessed 1 Feb 2020
15. H. Jagode, A. Danalis, H. Anzt, J. Dongarra, PAPI software-defined events for in-depth performance analysis. *Int. J. High Perform. Comput. Appl.* **33**, 1113–1127 (2019)
16. D. Kuzmin, J. Hämäläinen, *Finite Element Methods for Computational Fluid Dynamics: A Practical Guide* (SIAM, 2014)
17. I.L. Markov, A. Fatima, S.V. Isako, S. Boixo, Quantum supremacy is both closer and farther than it appears. Preprint at <https://arxiv.org/pdf/1807.10749> (2018). Accessed 15 Jan 2020
18. A. Messiah, *Quantum Mechanics*. Trans. by G. H. Temmer and J. Potter. Dover reprint, 1999 (1958)
19. M.A. Nielsen, I.L. Chuang, *Quantum Computation and Quantum Information*. 10th Anniversary Edition. (Cambridge, 2010)
20. P.D.T. O’Connor, *Practical Reliability Engineering*, 4th edn. (Wiley, 2002)
21. R. Patton, *Software Testing*, 2nd edn. (Sams Publishing, Indianapolis, 2005)
22. B. Russell, A.N. Whitehead, *Principia Mathematica*, vol I (Merchant Books, 1910)
23. W.C. Salmon, *The Foundations of Scientific Inference* (University of Pittsburgh Press, 1966)
24. J.F. Symons, J.K. Horner, Software intensive science. *Philos. Technol.* **27**, 461–477 (2014)

25. J.F. Symons, J.K. Horner, Software error as a limit to inquiry for finite agents: Challenges for the post-human scientist, in *Philosophy and Computing: Essays in Epistemology, Philosophy of Mind, Logic, and Ethics*, ed. by T. M. Powers, (Springer, New York, 2017), pp. 85–98
26. J.F. Symons, J.K. Horner, Why there is no general solution to the problem of software verification. *Found. Sci.* (2019). <https://doi.org/10.1007/s10699-019-09611-w>
27. A. Valmari, The state explosion problem, in *Lectures on Petri Nets I: Basic Models*, Lectures in Computer Science 1491, (Springer, 1998), pp. 429–528

Chess Is Primitive Recursive



Vladimir A. Kulyukin

1 Introduction

A deterministic two-player board game is played by two players on a finite board. The players take turns in choosing a move out of finitely many moves. Such a game has a unique starting board and end boards can be classified as a win for either player or a draw. Examples of deterministic two-player board games are Tic Tac Toe [1] and its many variants (e.g., Qubic [2]), chess, and checkers. Let an epoch be the set of all boards reachable from the starting board after a given number of moves. A deterministic two-player game is primitive recursive if there exists a primitive recursive (p.r.) function $G(p, i, j)$, where p is a player and i and j are epoch numbers such that $i < j$, that returns for p an optimal sequence of moves from a given board in epoch i to a board in epoch j if it is p 's turn to play at epoch i .

In a previous paper [3], we showed Tic Tac Toe to be p.r. In this paper, a proof is presented to show that chess is a deterministic p.r. game. In Sect. 2, several operators on Gödel numbers are defined. Section 3 presents a proof that chess is p.r. Conclusions are presented in Sect. 4.

2 Gödel Number Operators

All variables such as x, y, z, a, b, i, j, k , and t refer to natural numbers (i.e., elements of \mathbb{N}). The Greek letters α, γ , and ω with appropriate subscripts refer to auxiliary

V. A. Kulyukin (✉)

Department of Computer Science, Utah State University, Logan, UT, USA
e-mail: vladimir.kulyukin@usu.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_30

421

functions. All functions map \mathbb{N} to \mathbb{N} . Let $\langle x, y \rangle = z$, where $z = 2^x(2y + 1) \div 1$, $x = \max\{2^d \mid (z + 1)\}$, and $y = ((z + 1)/2^x - 1)/2$. The pairing functions $l(z)$ and $r(z)$ are defined in (1) and shown to be p.r. in [4], where \min is the minimalization function that returns the smallest natural number for which the predicate being minimalized is true.

$$\begin{aligned} l(z) &= \min_{x \leq z} \{(\exists y)_{\leq z} \{z = \langle x, y \rangle\}\} \\ r(z) &= \min_{y \leq z} \{(\exists x)_{\leq z} \{z = \langle x, y \rangle\}\} \end{aligned} \tag{1}$$

Let (a_1, \dots, a_n) be a sequence of numbers. The Gödel number of this sequence is defined in (2).

$$[a_1, \dots, a_n] = \prod_{i=1}^n p_i^{a_i} \tag{2}$$

The function $[a_1, \dots, a_n]$ is p.r., because $x \cdot y$, x^y , and p_i are p.r., as shown in [4]. The Gödel number of () is 1. Let $x = [a_1, \dots, a_n]$. Then the function $(x)_i = a_i$, $1 \leq i \leq n$ defined in (3) is shown to be p.r. in [4].

$$(x)_i = \min_{t \leq x} \{\neg(p_i^{t+1} \mid x)\}. \tag{3}$$

The length of x is the position of the last non-zero prime power in the Gödel representation of x defined by the p.r. function $Lt(x)$ in (4).

$$Lt(x) = \min_{i \leq x} \{(x)_i \neq 0 \wedge (\forall j)_{\leq x} \{j \leq i \vee (x)_j = 0\}\} \tag{4}$$

Let the p.r. function $\lfloor x/y \rfloor$ return the integer part of the quotient x/y [4]. Let $\gamma_1(i, b) \equiv i > Lt(b) \vee i < 1 \vee b < 1$. The p.r. function set in (5) sets the i -th element of b to x .

$$\text{set}(b, i, x) = \begin{cases} 0 & \text{if } \gamma_1(i, b), \\ \lfloor \frac{b}{p_i^{(b)_i}} \rfloor \cdot p_i^x & \text{otherwise.} \end{cases} \tag{5}$$

Let

$$\begin{aligned} \text{cntx}(x, y, 0) &= 0, \\ \text{cntx}(x, y, t + 1) &= \gamma_2(x, y, t, \text{cntx}(x, y, t)), \end{aligned}$$

where $s(x) = x + 1$ and

$$\gamma_2(x, y, t, c) = \begin{cases} 1 + c & \text{if } (y)_{s(t)} = x, \\ c & \text{otherwise.} \end{cases}$$

The p.r. function `count` in (6) returns the count of occurrences of x in y . Let $\text{in}(x, y) \equiv \text{count}(x, y) \neq 0$. In the remainder of the paper, $\text{in}(x, y)$ and $x \in y$ are used interchangeably.

$$\text{count}(x, y) = \text{cntx}(x, y, Lt(y)) \tag{6}$$

The p.r. function `rap` in (7) appends its first argument x to the right of the second argument y .

$$\text{rap}(x, y) = \begin{cases} [x] & \text{if } y = 0 \vee y = 1, \\ y \cdot P_{Lt(y)+1}^x & \text{otherwise.} \end{cases} \tag{7}$$

Let

$$\begin{aligned} \text{lcx}(x_1, x_2, 0) &= x_2, \\ \text{lcx}(x_1, x_2, t + 1) &= \gamma_3(x_1, t, \text{lcx}(x, y, t)), \end{aligned}$$

where $\gamma_3(x, t, y) = \text{rap}((x)_{s(t)}, y)$. The p.r. function \otimes_l in (8) places all elements of x_2 , in order, to the left of the first element of x_1 . Let $\otimes_l|_{i=1}^k x_i = x_1 \otimes_l x_2 \otimes_l \dots \otimes_l x_k = (\dots((x_1 \otimes_l x_2) \otimes_l \dots \otimes_l x_k) \dots) = (\dots(x_1 \otimes_l (x_2 \otimes_l (\dots \otimes_l (x_{k-1} \otimes_l x_k) \dots))) \dots)$.

$$x_1 \otimes_l x_2 = \text{lc}(x_1, x_2) = \text{lcx}(x_1, x_2, Lt(x_1)) \tag{8}$$

The p.r. function \otimes_r in (9) places all elements of x_2 , in order, to the right of the last element in x_1 . Let $\otimes_r|_{i=1}^k x_i = x_1 \otimes_r x_2 \otimes_r \dots \otimes_r x_k = (\dots((x_1 \otimes_r x_2) \otimes_r \dots \otimes_r x_k) \dots) = (\dots(x_1 \otimes_r (x_2 \otimes_r (\dots \otimes_r (x_{k-1} \otimes_r x_k) \dots))) \dots)$. Note that $0 \otimes_l x = 1 \otimes_l x = 0 \otimes_r x = 1 \otimes_r x = x$.

$$x_1 \otimes_r x_2 = \text{lc}(x_2, x_1) \tag{9}$$

Let $\text{rmx}(x, y, 0) = []$ and $\text{rmx}(x, y, t+1) = \gamma_4(x, y, \text{rmx}(x, y, t), s(t))$, where

$$\gamma_4(x, y, z, i) = \begin{cases} z & \text{if } (y)_i = x, \\ [(y)_i] \otimes_l z & \text{otherwise.} \end{cases}$$

The p.r. function $\text{rm}(x, y) = \text{rmx}(x, y, Lt(y))$ removes all occurrences of x from y .

Let $f(x)$ be a p.r. predicate and let $\text{mapx}_f(y, 0) = []$ and $\text{mapx}_f(y, t + 1) = \gamma_5(y, \text{mapx}_f(y, t), s(t))$, where

$$\gamma_5(y, z, i) = \begin{cases} z & \text{if } f((x)_i) = 0, \\ [(y)_i] \otimes_l z & \text{if } f((x)_i) = 1. \end{cases}$$

The p.r. function $\text{map}_f(y) = \text{map}_{x_f}(y, Lt(y))$ returns the list of occurrences of those elements x in y for which $f(x) = 1$.

Let $\text{pssx}(x, y, 0) = []$ and $\text{pssx}(x, y, t + 1) = \gamma_6(x, y, \text{pssx}(x, y, t), s(t))$, where

$$\gamma_6(x, y, z, i) = \begin{cases} [i] \otimes_l z & \text{if } (y)_i = x, \\ z & \text{otherwise.} \end{cases}$$

The p.r. function $\text{pstn}(x, y) = \text{pssx}(x, y, Lt(y))$ returns all positions of x in y .

3 Chess

We can encode a chess board as a Gödel number B with 64 elements (see Fig. 1). An empty cell is encoded as 1. A white pawn is encoded as $2 \leq n \leq 9$, the two white rooks are 10 and 17, the two white knights are 11 and 16, the two white bishops are 12 and 15, the white queen is 13, and the white king is 14. A black pawn is $18 \leq n \leq 25$, the two black rooks are 26 and 33, the two black knights are 27 and



26	27	28	29	30	31	1	33
18	19	20	21	1	23	24	25
1	1	1	1	1	32	1	1
1	1	1	1	22	1	1	1
1	1	1	1	6	1	1	1
1	1	1	1	1	16	1	1
2	3	4	5	1	7	8	9
10	11	12	13	14	15	1	17

Fig. 1 Chess board after 2 moves (above) and its Gödel number represented as a 2D matrix (below)

32, the two black bishops are 28 and 31, the black queen is 29, and the black king is 30. Let b_0 be the starting board. Then $(b_0)_j$, $1 \leq j \leq 16$, encode the black pieces, $(b_0)_j$, $49 \leq j \leq 64$, encode the white pieces, and $(b_0)_j$, $17 \leq j \leq 48$, encode the four empty rows in the middle of the board.

Let b be a board. The p.r. predicate $\gamma_7(b) \equiv \text{count}(14, b) = \text{count}(30, b) = 1$ ensures that b has exactly one white king ($i = 14$) and exactly one black king ($i = 30$). The predicate $\gamma_8(b) \equiv (\forall i)_{\leq 64} \{i \leq 1 \vee i = 14 \vee i = 30\} \vee \{\text{count}(i, b) \leq 1\}$ ensures that, unless i encodes an empty square ($i = 1$), the white king ($i = 14$), or the black king ($i = 30$), its count on b is 0 or 1. The predicate $\gamma_9(b) \equiv (\forall i)_{\leq 64} \{i \neq 1\} \vee \{32 \leq \text{count}(i, b) \leq 62\}$ ensures that the count of the empty spaces on b is between 32 (in the starting chess board) and 62 (when only the two kings remain on the board). The predicate $\text{valid}(b)$ in (10) is true if b is a valid chess board.

$$\text{valid}(b) \equiv \text{Lt}(b) = 64 \wedge \gamma_7(b) \wedge \gamma_8(b) \wedge \gamma_9(b) \quad (10)$$

For each piece x in a specific position, there is a set of positions reachable for x from that position. Let $z = [1, 2, \dots, 64]$ be the Gödel number encoding the board positions, where 1 encodes the top-left corner of the board and 64 encodes the bottom-right corner of the board, and let L_j^k be the Gödel number whose elements are the board positions where chess piece j can move from position k . For example, L_{15}^1 is the list of positions reachable by the light-colored bishop 15 from position 1. The lists of positions reachable by bishop 15 from the positions along the main diagonal are

$$\begin{aligned} L_{15}^1 &= [10, 19, 28, 37, 46, 55, 64]; \\ L_{15}^{10} &= [1, 3, 17, 19, 28, 37, 46, 55, 64]; \\ L_{15}^{19} &= [1, 10, 5, 12, 26, 33, 28, 37, 46, 55, 64]; \\ L_{15}^{28} &= [1, 10, 19, 7, 14, 21, 35, 42, 49, 37, 46, 55, 64]; \\ L_{15}^{37} &= [1, 10, 19, 28, 16, 23, 30, 44, 51, 58, 46, 55, 64]; \\ L_{15}^{46} &= [1, 10, 19, 28, 37, 32, 39, 53, 60, 55, 64]; \\ L_{15}^{55} &= [1, 10, 19, 28, 37, 46, 48, 62, 64]; \\ L_{15}^{64} &= [1, 10, 19, 28, 37, 46, 55]. \end{aligned}$$

Let W be the Gödel number whose elements are the white-colored cells on the chess board. Then L_{15} in (11) defines the Gödel number whose elements are all possible board positions for bishop 15.

$$L_{15} = \otimes_r |_{i \in W} L_{15}^i. \quad (11)$$

Such lists (i.e., Gödel numbers) can be computed in a p.r. fashion for all pieces and all positions on the board. Let $L = \otimes_r |_{i=2}^{33} [<i, L_i >]$ be the list of pairs $<i, L_i >$, where i denotes a chess piece and L_i is the list of all possible positions where i can move. The p.r. function in (12) returns, for each piece x at position i on b , all potentially reachable positions where x can move from i .

$$\text{prp}(x, i, b) = \begin{cases} r((L_x)_{\gamma_{10}(i, L_x)}) & \text{if } \text{valid}(b) \\ 0 & \text{otherwise,} \end{cases} \quad (12)$$

where $\gamma_{10}(i, L) = \min_{k \leq Lt(L)} \{i = l((L)_k)\}$ and $L_x = r((L)_{\gamma_{10}(x, L)})$.

The p.r. predicate $\text{wp}(x) \equiv 2 \leq x \leq 17$ is true if x is a white piece and the p.r. predicate $\text{bp}(x) \equiv 18 \leq x \leq 33$ is true if x is a black piece. Let the predicate $\text{arp}(x, i, b)$ be true if and only if position i on b is actually reachable for piece x (i.e., $\text{prp}(x, i, b) = 1$ and i is not blocked by another piece). Since $\text{arp}(x, i, b)$ can be defined in this manner by cases, each of which is defined in terms of p.r. predicates such as bp , wp , valid , Lt , rap , etc., and combinations of compositions and primitive recursions thereof, arp is p.r.

Let $z = [1, 2, \dots, 64]$ be the Gödel number encoding the board positions and b a chess board. Let $f(x) \equiv \text{arp}(x, j, b)$, where $j \in z$ and $\text{wp}(x) = 1$ or $\text{bp}(x) = 1$. Then the p.r. function $\text{alst}(x, b) = \text{map}_f(x, z)$ returns the Gödel number of actually reachable positions for x on b .

Let $\text{bkp}(b) = (\text{pstrn}(30, b))_1$ be a p.r. function that returns the position of the black king on b . The black king is checked when there is a white piece (other than the white king encoded as 14) for which the current position of the black king is actually reachable. Formally, $\text{bchk}(b) \equiv (\exists x)_{<34} \{\text{wp}(x) \wedge x \neq 14 \wedge \gamma_{11}(x, b)\}$, where $\gamma_{11}(x, b) \equiv (\exists j)_{<65} \{j > 0 \wedge \alpha(x, j, b)\}$ and $\alpha(x, j, b) \equiv (\text{pstrn}(x, b))_1 = j \wedge \text{in}(\text{bkp}(b), \text{alst}(x, b))$. The black king is mated if, when checked, it cannot move to any cell that is not actually reachable by a white piece. The p.r. predicate bmtd in (13) defines this logic.

$$\text{bmtd}(b) \equiv \text{bchk}(b) \wedge \gamma_{12}(b), \quad (13)$$

where $\gamma_{12}(b) \equiv (\forall j)_{<65} \{\neg \text{in}(j, \text{alst}(30, b)) \vee \gamma_{13}(j, b)\}$ and $\gamma_{13}(j, b) \equiv (\exists x)_{<34} \{\text{wp}(x) \wedge \text{in}(j, \text{alst}(x, b))\}$. The same logic can be used to define a p.r. predicate $\text{wmtd}(b)$ to return 1 when the white king is mated and 0, otherwise.

A draw by stalemate occurs when the player whose turn it is to move is not in check but has no legal move. The p.r. black stalemate predicate $\text{bstlmt}(b) \equiv \neg \text{bchk}(b) \wedge \gamma_{14}(b)$, where $\gamma_{14}(b) \equiv (\forall j)_{<65} [\neg \text{bp}(j) \vee \text{Lt}(\text{alst}(j, b)) = 0]$, checks if the black king is not checked and no black piece has actually reachable positions. A white stalemate can be defined in the same fashion. All cases of the dead position rule can be defined as p.r. predicates with count , valid , in , $=$, and boolean combinations thereof.

A draw by repetition is achieved when the same position occurs three times in a row with the same player to move. The 50-move rule states that a game is a draw when the last 50 moves contain no capture or pawn move. The dead position rule applies when neither player can checkmate the opponent by any series of moves. The dead position rule applies to situations when there are only two kings left on the board, when one side has the king and a bishop and the other side has the king, when one side has the king and a knight and the other side has the king, when both

sides have the king and a bishop and the bishops are both light-colored or dark-colored. The rules for the draw by repetition and the 50-move rule will be outlined below after we formalize the notion of the board history. Consider the p.r. function in (14).

$$(b)_j^x = \begin{cases} \text{set}(b, j, x) & \text{if } \gamma_{15}(x, b) \\ 0 & \text{otherwise,} \end{cases} \tag{14}$$

where $\gamma_{15}(x, b) \equiv \text{valid}(b) \wedge \{\text{wp}(x) \vee \text{bp}(x)\} \wedge \text{in}(j, \text{alst}(x, b))$. For example, the chess board in Fig. 1 is

$$((((b_0)_{37}^6)_{29}^{22})_{46}^{16})_{22}^{32}$$

Let b^1 be the list of all possible boards obtained from b by exactly one move of the white player, assuming that it is the white's turn to move.

$$b^1 = \otimes_r |_{x=2}^{17} \otimes_r |_{j=1}^{64} [(b)_j^x].$$

Let b^2 be the list of all possible moves obtained from b by exactly one move of the black player, assuming that it is the black's turn to move.

$$b^2 = \otimes_r |_{x=18}^{33} \otimes_r |_{j=1}^{64} [(b)_j^x].$$

Since both b^1 and b^2 are p.r., we can combine b^1 and b^2 into a single p.r. function $\text{pm}(x, p)$ that maps the current board to the list of all possible boards obtained from it by exactly one move of either player.

$$\text{pm}(b, p) = \begin{cases} b^1 & \text{if } p = 1 \wedge \text{valid}(b) \\ b^2 & \text{if } p = 2 \wedge \text{valid}(b) \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

Let $Z = [b_{i_0}, b_{i_1}, \dots, b_{i_k}]$ such that $\text{valid}(b_{i_j}) = 1, 0 \leq j \leq k$. Let $p \in \{1, 2\}$. The p.r. function $\text{scr}(Z, p)$ in (16) takes a Gödel number that consists of valid boards and a player's number and returns another Gödel number that consists of successor boards such that each successor board is obtained from one of the boards in Z by exactly one move of p , assuming that it is p 's turn to move.

$$\text{scr}(Z, p) = \text{rm}(0, \otimes_r |_{i=1}^{\text{Lt}(Z)} \text{pm}((Z)_i, p)) \tag{16}$$

Let b_1 and b_2 be two boards and let $p \in \{1, 2\}$ be a player whose turn it is to play on b_1 . The p.r. predicate $\text{prn}(p, b_1, b_2) \equiv \text{valid}(b_2) \wedge \text{in}(b_2, \text{pm}(p, b_1))$ is true when b_2 is in the Gödel number of the boards obtained from b_1 by exactly one

move of p on it. In other words, b_1 is the parent of b_2 . Let the p.r. function $B(t)$, defined in (17), return the Gödel number that includes all boards, actually reachable from b_0 after t moves. We will refer to each $B(t)$ as *epoch* t . Let $B(0) = \llbracket [b_0], 1 \rrbracket$.

$$B(t + 1) = \llbracket \text{scr}(l((B(t))_1), \gamma_{16}(s(t))) \rrbracket, \quad (17)$$

where $\gamma_{16}(x) = 1$ if $\neg(2|x)$ and 2 if $2|x$. Let $G_0 = [b_0]$ and G_i , for $i > 0$, be the Gödel number encoding all boards actually reachable from the boards in G_{i-1} in 1 move by the appropriate player. Let $b \in G_i$, $G_{i-1} = l((B(i-1))_1)$, and $p = r((B(i-1))_1)$. The p.r. function ipb in (18) returns the index of the parent of b in G_{i-1} for $i > 0$.

$$\text{ipb}(b, i) = \min_{t \leq L^i(G_{i-1})} \{\text{prn}(p, (G_{i-1})_t, b)\} \quad (18)$$

The p.r. function prb in (19) returns the parent of b .

$$\text{prb}(b, i) = (l((B(i-1))_1))_{\text{ipb}(b, i)} \quad (19)$$

The p.r. function prbs in (20) computes the Gödel number whose last element is b and whose previous elements are its predecessors. In other words, element 8 is the parent board of element 9, element 7 is the parent board of element 8, etc.

$$\begin{aligned} \text{prbs}(b, i) = & \llbracket \text{prb}(\dots(\text{prb}(b, i), 7), \dots, 1), \\ & \dots, \\ & \text{prb}(\text{prb}(\text{prb}(b, i), 7), 6), \\ & \text{prb}(\text{prb}(b, i), 7), \text{prb}(b, i), b \rrbracket \quad (20) \end{aligned}$$

Let $X = \text{prbs}(b, i)$ such that $i > 7$ and $\text{valid}(b) = 1$. The p.r. predicate $\text{drw3r}(b, i) \equiv \gamma_{17}(X)$, where $\gamma_{17}(X) \equiv \{(X)_1 = (X)_5 = (X)_9\} \wedge \{(X)_2 = (X)_6\} \wedge \{(X)_3 = (X)_7\} \wedge \{(X)_4 = (X)_8\}$ is true if b is a threefold repetition board. To put it differently, in the list of b 's predecessors, elements 1 and 5 must be the same as b (i.e., element 9), element 2 must be the same as element 6, element 3 is the same as element 7, and element 4 is the same as element 8.

It is straightforward to extend the definition of $\text{prbs}(b, i)$ to a p.r. predicate $\text{drw50}(b, i)$ that computes 49 predecessors of a valid board b in epoch $B(i)$ and checks if each board in the Gödel number of the predecessors and b itself contains no capture, which can be done by comparing the number of pieces on a given board and its immediate predecessor (i.e., its parent), or a pawn move, which can be done by comparing the pawn positions of all the predecessor boards of b and b itself. All these functions are p.r., because they manipulate Gödel numbers.

We can similarly express, in a p.r. fashion, each case of the dead position rule. For example, checking if a given board b contains only two kings or whether the white has the king and a knight and the black has only the king is p.r., because it

requires checking p.r. properties of a given Gödel number. Consequently, we may assume that there is a p.r. predicate $\text{draw}(b, i)$ that returns 1 if a valid board b in epoch $i > 0$ is a draw and 0 otherwise.

Let $t \in \mathbb{N}$. We define the chess game history in (21) as the Gödel number encoding the boards at each epoch and the player whose turn it is to play at the next epoch.

$$H(t) = \otimes_r |_{i=0}^t B(i) \quad (21)$$

For example $H(3) = [B(0), B(1), B(2), B(3)] = [\langle G_0, 1 \rangle, \langle G_1, 2 \rangle, \langle G_2, 1 \rangle, \langle G_3, 2 \rangle]$. Let $t, i, j \in \mathbb{N}$. Let $G_i^t = l((H(t))_i)$, $b_{i,j}^t = (G_i^t)_j$, $L_t = \text{Lt}(H(t))$, and $L_i^t = \text{Lt}(G_i^t)$. The p.r. predicate $\text{ww}(t)$, $t \geq 0$, in (22) returns true if there is a board in epoch i , $0 \leq i \leq t$, where the white checkmates its opponent.

$$\text{ww}(t) \equiv (\exists i)_{\leq L_t} \{(\exists j)_{\leq L_i^t} \{\text{bmt}d(b_{i,j}^t)\}\} \quad (22)$$

The p.r. predicate $\text{bw}(t)$ in (23) is true if there is a board in epoch i , $0 \leq i \leq t$, where the black checkmates its opponent.

$$\text{bw}(t) \equiv (\exists i)_{\leq L_t} \{(\exists j)_{\leq L_i^t} \{\text{wmt}d(b_{i,j}^t)\}\} \quad (23)$$

If $H(t)$ is the history of the game, then the white can win only in the even-numbered epochs and the black can win only in the odd-numbered epochs. We can define two predicates $W_w(m)$ and $W_b(m)$ that are true if the white or black, respectively, wins within t moves. Specifically, $W_w(m) \equiv (\exists t)_{\leq m} \{\text{ww}(t)\}$ and $W_b(m) \equiv (\exists t)_{\leq m} \{\text{bw}(t)\}$. The p.r. predicate in (24) combines both predicates into one.

$$W(p, m) = \begin{cases} W_w(m) & \text{if } p = 1 \\ W_b(m) & \text{if } p = 2 \\ 0 & \text{otherwise.} \end{cases} \quad (24)$$

In a similar fashion, we can define the p.r. predicate in (25) that is true if a draw is achieved for player p within m moves.

$$D(p, m) \equiv (\exists i)_{\leq L_t} \{(\exists j)_{\leq L_i^t} \{\text{draw}(b_{i,j}^t, i)\}\}. \quad (25)$$

Let $p \in \{1, 2\}$, $t \in \mathbb{N}$. The p.r. functions $W_{\leq}(p, t)$ in (26) and $D_{\leq}(p, t)$ in (27) return the lists of all win and draw boards for p , respectively, within t moves.

$$W_{\leq}(p, t) = \otimes_r |_{i=1}^t W(p, i) \quad (26)$$

$$D_{\leq}(p, t) = \otimes_r |_{i=1}^t D(p, i) \quad (27)$$

Let $b_x \in l((B(i))_1)$ and $b_y \in l((B(j))_1)$, where $i < j$. Let

$$\begin{aligned} \text{ptx}(b_x, b_y, 0) &= \gamma_{18}(b_x, b_y), \\ \text{ptx}(b_x, b_y, t + 1) &= [b_y] \otimes_l \text{ptx}(b_x, \text{prb}(b_y, s(t)), t), \end{aligned}$$

where

$$\gamma_{18}(x, y) = \begin{cases} [x] & \text{if } x = y, \\ 0 & \text{otherwise.} \end{cases}$$

The p.r. function path in (28) gives the list of boards, possibly empty, from the board $b_x \in B(i)$ to the board $b_y \in B(j)$.

$$\text{path}(b_x, b_y, i, j) = \text{ptx}(b_x, b_y, j - i) \quad (28)$$

If $\text{path}(b_x, b_y, i, j) \neq 0$, $b_y \in B(j)$ is *reachable* from $b_x \in B(i)$. If $\text{path}(b_x, b_y, 1, 3) = 0$, then b_y is *unreachable* from b_x . Let $b \in l((B(i))_1)$, $k \geq 0$, $i < j$, and let $Z \in \mathbb{N}$ be a list of boards. Let

$$\begin{aligned} \text{ppx}(b, Z, i, j, 0) &= [], \\ \text{ppx}(b, Z, i, j, t + 1) &= \gamma_{19}(b, Z, t, i, j) \otimes_l \\ &\quad \text{ppx}(b, Z, i, j, t), \end{aligned}$$

where $\gamma_{19}(b, Z, t, i, j) = [\text{path}(b, (Z)_{s(t)}, i, j)]$. The p.r. function ppx returns a list of paths from a given board b to each board in Z . Let $\text{ppxx}(b, Z, i, j) = \text{ppx}(b, Z, i, j, Lt(Z))$. The p.r. function in (29) returns the list of all paths, possibly empty, from $b \in l((B(i))_1)$ to a win board $b' \in l((B(j))_1)$.

$$\text{wpss}(b, i, j) = \otimes_r |_{k=i+1}^j \gamma_{20}(b, k, i, j), \quad (29)$$

where $\gamma_{22}(b, k, i, j) = \text{ppxx}(b, W(p, k), i, j)$, where $p = 1$ if $2|i$ and $p = 2$, otherwise. The p.r. function in (30) removes all empty paths from the list returned by wpss .

$$\text{wps}(b, i, j) = \text{rm}(0, \text{wpss}(b, i, j)) \quad (30)$$

The p.r. function in (31) returns the list of all paths, possibly empty, that start at $b \in l((B(i))_1)$ and end with a draw board $b' \in l((B(j))_1)$.

$$\text{dpss}(b, i, j) = \otimes_r |_{k=i+1}^j \gamma_{21}(b, k, i, j), \quad (31)$$

where $\gamma_{21}(b, k, i, j) = \text{ppxx}(b, D(p, k), i, j)$, where $p = 1$ if $2|i$ and $p = 2$, otherwise. The p.r. function in (32) removes all empty paths from the list returned by dpss .

$$\text{dps}(b, i, j) = \text{rm}(0, \text{dpsS}(b, i, j)) \quad (32)$$

Let $0 \leq k < t$, $k < j \leq t$, $p \in \{1, 2\}$, and $b \in l((\mathbb{B}(k))_1)$. If $p = 1$ (i.e., p plays white), then p is the max player whose objective is to maximize the utility score of b . Let the highest utility score that can be assigned to b be 3 if there is at least one win board $b' \in \mathbb{B}(j)$ reachable from b . If there are no reachable win boards, let the utility of b be 2 so long as there is at least one draw board $b' \in \mathbb{B}(j)$ reachable from b . Let b have the lowest utility score of 1 when there is no win or draw board $b' \in \mathbb{B}(j)$ reachable from b . Let the utility score of 0 be assigned to invalid boards. The p.r. function in (33) returns the utility score of b for $p = 1$, where $\gamma_{22}(b, k, t) \equiv Lt(\text{dps}(b, k, t)) = Lt(\text{wps}(b, k, t)) = 0$.

$$U_{\max}(b, k, t) = \begin{cases} 3 & \text{if } Lt(\text{wps}(b, k, t)) > 0, \\ 2 & \text{if } Lt(\text{dps}(b, k, t)) > 0, \\ 1 & \text{if } \gamma_{22}(b, k, t), \\ 0 & \text{if } \neg\text{valid}(b). \end{cases} \quad (33)$$

If $p = 2$ (i.e., p plays black), then p is the min player whose objective is to minimize the utility score of b . Let the utility score of b be 1 if there is at least one win board $b' \in \mathbb{B}(j)$ reachable from b . If there are no win boards in $\mathbb{B}(j)$ reachable from b , let the utility score of b be 2 so long as there is at least one draw board in $b' \in \mathbb{B}(j)$ reachable from b . Let the highest utility score of 3 indicate that there is no win or draw board $b' \in \mathbb{B}(j)$ reachable from b . Again, let the utility score of 0 be assigned to invalid boards. The p.r. function in (34) returns the utility score for $p = 2$ for player 0.

$$U_{\min}(b, k, t) = \begin{cases} 1 & \text{if } Lt(\text{wps}(b, k, t)) > 0, \\ 2 & \text{if } Lt(\text{dps}(b, k, t)) > 0, \\ 3 & \text{if } \gamma_{22}(b, k, t), \\ 0 & \text{if } \neg\text{valid}(b). \end{cases} \quad (34)$$

Let $b \in l((\mathbb{B}(i))_1)$, $0 \leq i < j$. The p.r. function U in (35) returns the utility score of $b \in \mathbb{B}(i)$ for player $p \in \{1, 2\}$ when the game continues from epoch $\mathbb{B}(i)$ to epoch $\mathbb{B}(j)$.

$$U(b, p, i, j) = \begin{cases} U_{\max}(b, i, j) & \text{if } 2|i, \\ U_{\min}(b, i, j) & \text{if } \neg(2|i), \\ 0 & \text{otherwise.} \end{cases} \quad (35)$$

Let $0 \leq i < j$. The p.r. function fb in (36) returns a list of boards with a given utility score $x \in \{1, 2, 3\}$ for player p from epoch i to epoch j .

$$\text{fb}(p, i, j, x) = \text{fbx}(p, i, j, Z, L, x), \quad (36)$$

where $Z = l((B(i))_1)$, $L = Lt(l((B(j))_1))$, $\text{fbx}(p, i, j, Z, 0, x) = []$, $\text{fbx}(p, i, j, Z, t+1, x) = \gamma_{23}(p, i, j, (Z)_{s(t)}, \text{fbx}(p, i, j, Z, t, x), x)$, and

$$\gamma_{23}(p, i, j, b, Z, x) = \begin{cases} [b] \otimes_l Z & \text{if } \cup(b, p, i, j) = x, \\ b & \text{if } \cup(b, p, i, j) \neq x. \end{cases}$$

Let $0 \leq i < j$ and $\alpha_3(p, i, j) \equiv \text{fb}(p, i, j, 3) \neq 0$, $\alpha_2(p, i, j) \equiv \text{fb}(p, i, j, 2) \neq 0 \wedge \text{fb}(p, i, j, 3) = 0$, and $\alpha_1(p, i, j) \equiv \text{fb}(p, i, j, 1) \neq 0 \wedge \text{fb}(p, i, j, 3) = \text{fb}(p, i, j, 2) = 0$. The p.r. function fb_{max} returns a list of boards for player $p = 1$ from epoch i to epoch j .

$$\text{fb}_{max}(p, i, j) = \begin{cases} \text{fb}(p, i, j, 3) & \text{if } \alpha_3(p, i, j), \\ \text{fb}(p, i, j, 2) & \text{if } \alpha_2(p, i, j), \\ \text{fb}(p, i, j, 1) & \text{if } \alpha_1(p, i, j), \\ 0 & \text{otherwise.} \end{cases}$$

Let $\beta_1(p, i, j) \equiv \text{fb}(p, i, j, 1) \neq 0$, $\beta_2(p, i, j) \equiv \text{fb}(p, i, j, 2) \neq 0 \wedge \text{fb}(p, i, j, 1) = 0$, and $\beta_3(p, i, j) \equiv \text{fb}(p, i, j, 3) \neq 0 \wedge \text{fb}(p, i, j, 1) = \text{fb}(p, i, j, 2) = 0$. The p.r. function fb_{min} returns a list of boards for player $p = 2$ from epoch i to epoch j .

$$\text{fb}_{min}(p, i, j) = \begin{cases} \text{fb}(p, i, j, 1) & \text{if } \beta_1(p, i, j), \\ \text{fb}(p, i, j, 2) & \text{if } \beta_2(p, i, j), \\ \text{fb}(p, i, j, 3) & \text{if } \beta_3(p, i, j), \\ 0 & \text{otherwise.} \end{cases}$$

The p.r. function fb_{mnx} in (37) returns the list of optimal boards for player $p \in \{1, 2\}$ from epoch i to epoch j .

$$\text{fb}_{mnx}(p, i, j) = \begin{cases} \text{fb}_{max}(p, i, j) & \text{if } p = 1, \\ \text{fb}_{min}(p, i, j) & \text{if } p = 2, \\ 0 & \text{otherwise.} \end{cases} \quad (37)$$

The p.r. function $\text{et}(k)$ in (38) determines whose turn it is to play at epoch $B(k)$.

$$\text{et}(k) = \begin{cases} 1 & \text{if } 2|k, \\ 2 & \text{if } \neg(2|k). \end{cases} \quad (38)$$

Let $0 \leq k < t$. Let

$$\omega_1(k, t) = \otimes_r |_{j=k}^{t-1} [(\text{fb}_{\text{mnx}}(\text{et}(j), j, t))_1].$$

The p.r. ω_1 function chooses the first board returned from the function fb_{mnx} . Other p.r. functions can also be defined to inspect the first n boards returned by fb_{mnx} . The p.r. function bseq in (39) returns a sequence of optimal boards for player p whose turn it is to play at k .

$$\text{bseq}(p, k, t) = \begin{cases} \omega(k, t) & \text{if } \text{et}(k) = p, \\ [] & \text{otherwise.} \end{cases} \quad (39)$$

Let $0 \leq k < t$ and let b_1 and b_2 be two boards such that $\phi(b_1, b_2, k) \equiv \text{in}(b_1, l((\mathbb{B}(k))_1)) \wedge \text{in}(b_2, l((\mathbb{B}(k+1))_1))$ and $\text{prb}(b_2, k+1) = b_1$. Let

$$\omega_2(b_1, b_2) = \min_{i \leq 64} \{(b_1)_i = 1 \wedge (b_2)_i \neq 1\}.$$

The p.r. function $\omega_3(b_1, b_2)$ extracts a move $\langle p, j \rangle$, where $1 \leq j \leq 64$, that changes b_1 to b_2 .

$$\omega_3(b_1, b_2, k) = \begin{cases} \langle \text{et}(k), \omega_2(b_1, b_2) \rangle & \text{if } \phi(b_1, b_2, k), \\ 0 & \text{otherwise.} \end{cases} \quad (40)$$

Let $Z = \text{bseq}(p, k, t) = [b_1, \dots, b_{t-k+1}]$ and let

$$\text{mseq}(p, k, t) = \otimes_r |_{i=1}^{L^t(Z)-1} [\omega_3((Z)_i, (Z)_{i+1}, k-1+i)]$$

Let $\gamma_{24}(p, i) \equiv \{2|i \wedge p = 1\} \vee \{\neg(2|i) \wedge p = 2\}$. The p.r. function in (41) defines a game of chess for player p and epoch i by returning a sequence of optimal moves for p beginning at epoch $\mathbb{B}(i)$ and ending at epoch $\mathbb{B}(j)$.

$$G(p, i, j) = \begin{cases} \text{mseq}(p, i, j) & \text{if } \gamma_{24}(p, i), \\ 0 & \text{otherwise.} \end{cases} \quad (41)$$

Since $G(i, j)$ is p.r., we have the following theorem.

Theorem *Chess is a deterministic two-player p.r. game.*

4 Conclusion

A proof is presented to show that chess is a deterministic two-player primitive recursive game. To the extent that the proof holds, chess can be characterized in terms of primitive recursive functions. If this is the case, some deterministic two-player games and processes that can be formalized as such are likely to have algorithmic solutions that outperform human players. The techniques developed in this paper may lead to proofs that other deterministic two-player board games are primitive recursive and contribute to the theory of primitive recursive functions [5].

References

1. T. Bolon, *How to Never Lose at Tic Tac Toe* (Book Country, New York, NY, USA, 2013)
2. W. Daly, Jr., Computer Strategies for the Game of Qubic, M. Eng. thesis, Department of Electrical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts, USA, Feb. 1961
3. V. Kulyukin, On primitive recursiveness of Tic Tac Toe, in *Proceedings of the International Conference on Foundations of Computer Science (FCS'19)*, pp. 9–15, Las Vegas, NV, USA, Jul. 29–Aug. 01, 2019
4. M. Davis, R. Sigal, E. Weyuker, *Computability, Complexity, and Languages: Fundamentals of Theoretical Computer Science*, 2nd edn. (Harcourt, Brace & Company, Boston, MA, USA, 1994)
5. H. Rogers, Jr., *Theory of Recursive Functions and Effective Computability* (The MIT Press, Cambridge, MA, USA, 1988)

How to Extend Single-Processor Approach to Explicitly Many-Processor Approach



János Végh 

1 Introduction

Physical implementations of a computer processor in the 70-year old computing paradigm have several limitations [1]. As the time passes, more and more issues come to light, but development of processor, the *central* element of a computer, could keep pace with the growing demand on computing till some point. Around 2005 it became evident that the price paid for keeping Single Processor Approach (SPA) paradigm [2], (as Amdahl coined the wording), became too high. “The *implicit hardware/software contract*, that increases transistor count and power dissipation, was OK as long as architects maintained the existing sequential programming model. This contract led to innovations that were inefficient in transistors and power—such as multiple instruction issue, deep pipelines, out-of-order execution, speculative execution, and prefetching—but which increased performance while preserving the sequential programming model” [3]. The conclusion was that “*new ways of exploiting the silicon real estate need to be explored*” [4].

“*Future growth in computing performance must come from parallelism*” [5] is the common point of view. However, “*when we start talking about parallelism and ease of use of truly parallel computers, we’re talking about a problem that’s as hard as any that computer science has faced*” [3]. Mainly because of this, parallel utilization of computers could not replace the energy-wasting solutions introduced

Projects no. 125547 and 135812 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K funding scheme. Also the support by project 101005020 from ERC is acknowledged.

J. Végh (✉)
Kalimános BT, Debrecen, Hungary

to the formerly favored single-thread processors. They remained in the Multi-Core and/or Many-Core (MC) processors, greatly contributing to their dissipation and, through this, to the overall crisis of computing [6].

Computing paradigm itself, the *implicit hardware/software contract*, was suspected even more explicitly: “*Processor and network architectures are making rapid progress with more and more cores being integrated into single processors and more and more machines getting connected with increasing bandwidth. Processors become heterogeneous and reconfigurable ... No current programming model is able to cope with this development, though, as they essentially still follow the classical von Neumann model*” [7]. On one side, when thinking about “advances beyond 2020”, the solution was expected from the “*more efficient implementation of the von Neumann architecture*” [8]. On the other side, there are statements such as “*The von Neumann architecture is fundamentally inefficient and non-scalable for representing massively interconnected neural networks*” [9].

In our other works [10–13] we have pointed out that one of the major reasons is neglecting the temporal behavior of computing components. The other major reason is, that the architecture developed for that classic paradigm is development-unaware, and cannot be equally good for the present needs and the modern paradigm. These two reasons together represent the major bottleneck—among others—to build supercomputers having reasonable efficiency in solving real-life tasks and biology-mimicking systems with the required size and efficiency, such as Artificial Intelligence (AIs) [14, 15] and brain simulators [16]. The interplay of these two reasons is that conventional processors do not have autonomous communication. The classic paradigm is about a segregated processor and, because of this, its communication is implemented using Input/Output (I/O) instructions and needs help of the operating system (OS). Both of these features increase non-payload (and sequential!) portion of the code and so they degrade efficiency, especially in excessive systems.

It is worth, therefore, to scrutinize that *implicit hardware/software contract*, whether the processor architecture could be adapted in a better way to the changes that occurred in the past seven decades in technology and utilization of computing. *Implicitly*, both hardware (HW) and software (SW) solutions advantageously use multi-processing. The paper shows that using a less rigid interpretation of terms that that contract is based upon, one can extend the single-thread paradigm to use several processors *explicitly* (enabling direct core-to-core interaction), without violating the ‘contract’, the 70-year old HW/SW interface.

Section 2 shortly summarizes some of the major challenges, modern computing is expected to cope with and sketches the principles that enable it to give a proper reply. The way to implement those uncommon principles proposed here is discussed in Sect. 3. Because of the limited space, only a few of the advantages are demonstrated in Sect. 4.

2 The General Principles of EMPA

During the past two decades, computing developed in direction to conquer also some extremes: the ‘ubiquitous computing’ led to billions of connected and interacting processors [17], the always higher need for more/finer details, more data and shorter processing times led to building computers comprising millions of processors to target challenging tasks [18], different cooperative solutions [19] attempt to handle the demand of dynamically varying computing in the present, more and more mobile, computing. *Using computing under those extreme conditions led to shocking and counter-intuitive experiences* that can be comprehended and accepted using parallels with modern science [10].

Developing a new computing paradigm being able to provide a theoretical basis for the state of the art of computing cannot be postponed anymore. Based on that, one must develop different types of processors. As was admitted following the failure of supercomputer Aurora’18: “*Knights Hill* was canceled and instead be replaced by a “new platform and new microarchitecture specifically designed for exascale”” [20]. Similarly, we expect shortly to admit that building large-scale AI systems is simply not possible based on the old paradigm and architectural principles [14, 15, 21]. The new architectures, however, require a new computing paradigm, that can give a proper reply to power consumption and performance issues of our present-day computing.

2.1 Overview of the Modern Paradigm

The new paradigm proposed here is based on fine distinctions in some points, present also in the old paradigm. Those points, however, must be scrutinized individually, whether and how long omissions can be made. These points are:

- consider that *not only one processor* (aka Central Processing Unit) exists, i.e.
 - processing capability is *one of the resources* rather than a central singleton
 - not necessarily *the same processing unit* is used to solve all parts of the problem
 - a kind of redundancy (an easy method of replacing a flawed processing unit) through using virtual processing units is provided (mainly to *increase the mean time between technical errors*)
 - instruction stream can be transferred to another processing unit [22, 23]
 - *different processors can and must cooperate* in solving a task, including direct data and control exchange between cores, communicating with each other, being able to set up ad-hoc assemblies for more efficient processing in a flexible way
 - the large number of processors can be used for *replacing memory operations with using more processors*
 - a core can outsource the received task

- misconception of segregated computer components is reinterpreted
 - *efficacy of using a vast number of processors is increased* by using multi-port memories (similar to [24])
 - a “memory only” concept (somewhat similar to that in [25]) is introduced (as opposed to the “registers only” concept), using *purpose-oriented, optionally distributed, partly local, memory banks*
 - principle of locality is introduced at hardware level, through introducing hierarchic buses
- misconception of “sequential only” execution [26] is reinterpreted
 - von Neumann required only “proper sequencing” for a single processing unit; this concept is *extended* to several processing units
 - tasks are broken into reasonably sized and logically interconnected fragments
 - the “one-processor-one process” principle remains valid for task fragments, but not necessarily for the complete task
 - fragments can be executed (at least partly) simultaneously if both data dependence and hardware availability enables it (another kind of asynchronous computing [27])
- a closer hardware/software cooperation is elaborated
 - hardware and software only exist together: the programmer works with virtual processors, in the same sense as [28] uses this term, and lets computing system to adapt itself to its task at run-time, through mapping virtual processors to physical cores
 - when a hardware has no duty, it can sleep (“does not exist”, does not take power)
 - the overwhelming part of the duties such as synchronization, scheduling of the OS are taken over by the hardware
 - the compiler helps work of the processor with compile-time information and the processor can adapt (configure) itself to its task depending on the actual hardware availability
 - strong support for multi-threading, resource sharing and low real-time latency is provided, at HW level
 - the internal latency of large-scale systems is much reduced, while their performance is considerably enhanced
 - task fragments shall be able to return control voluntarily without the intervention of OS, enabling to implement more effective and more simple operating systems
 - the processor becomes “green”: only working cores take power

2.2 Details of the Concept

We propose to work at programming level with *virtual processors* and to map them to physical cores at run-time, i.e., to *let the computing system to adapt itself to its task*. A major idea of EMPA is to use *quasi-thread (QT)* as atomic unit of processing, that comprises both *HW* (the physical core) and the *SW* (the code fragment running on the core). Its idea was derived with having in mind the best features of both *HW* core and *SW* thread. *QTs* have “*dual nature*” [10]: in the *HW* world of “classic computing” they are represented as a ‘core’, in *SW* world as a ‘thread’. However, they are the same entity in the sense of ‘modern computing’. We borrow the terms ‘core’ and ‘thread’ from conventional computing, but in ‘modern computing’, they can actually exist only together in a time-limited way.¹ *EMPA is a new computing paradigm* (for an early version see [29]) which needs a new underlying architecture, rather than a new kind of parallel processing running on a conventional architecture, so it can be reasonably compared to terms and ideas used in conventional computing only in a minimal way; although the new approach adapts many of its ideas and solutions, furthermore borrows its terms, from ‘classic computing’.

One can break the executable task into reasonably sized and loosely dependent Quasi-Thread (*QT*)s. (The *QTs* can optionally be nested, akin to subroutines.) In *EMPA*, for every new *QT* a new independent Processing Unit (*PU*) is also implied, the internals (*PC* and part of registers) are set up properly, and they execute their task independently² (but under the supervision of the processor comprising the cores).

In other words: *we consider processing capacity as a computing resource* in the same sense as memory is considered as a storage resource. This approach enables programmers to work with virtual processors (mapped to physical *PU*s by the computer at run-time) and they can utilize quick resource *PU*s to replace utilizing slow resource memory (say, renting a quick processor from a core pool can be competitive with saving and restoring registers in slow memory, for example when making a subroutine call). The third primary idea is that *PU*s can cooperate in various ways, including data and control synchronization, as well as *outsourcing part of the received job (received as an embedded QT)* to a helper core. An obvious example is to outsource housekeeping activity to a helper core: counting, addressing, comparing, can be done by a helper core, while the main calculation remains to the originally delegated core. As mapping to physical cores occurs at run-time (a

¹Akin to dynamic variables on the stack: their lifetime is limited to the period when the *HW* and *SW* are appropriately connected. The physical memory is always there, but it is “stack memory” only when handled adequately by the *HW/SW* components.

²Although the idea of executing the single-thread task “in pieces” may look strange for the first moment, the same happens when the OS schedules/blocks a task. The key differences are that in *EMPA* not the same processor is used, the Explicitly Many-Processor Approach (*EMPA*) cuts the task into fragments in a reasonable way (preventing issues like priority inversion [30]). The *QTs* can be processed at the same time as long as their mathematical dependence and the actual *HW* resource availability enable it.

about enhancing performance but has no information about actual run-time HW availability. Furthermore, it has no way to tell its findings to the processor. Processor has HW availability information but has to “reinvent the wheel” to enhance its performance; in real-time. In EMPA, compiler puts its findings in the executable code in form of meta-instructions (“configware”), and the actual core executes them with the assistance of a new control layer of the processor. The processor can choose from those options, considering actual HW availability, in a style ‘**if** NeededNumberOfResourcesAvailable **then** Method1 **else** Method2’, maybe nested one into another.

2.3 Some Advantages of EMPA

The approach results in several considerable advantages, but the page limit enables us to mention just a few of them.

- as a new *QT* receives a new PU, there is no need to save/restore registers and return address (less memory utilization and less instruction cycles)
- OS can receive its PU, initialized in kernel mode and can promptly (i.e., without the need of context change) service the requests from the requestor core
- for resource sharing, a PU can be temporarily delegated to protect the critical section; the next call to run the code fragment with the same offset shall be delayed (by the processor) until processing by the first PU terminates
- processor can natively accommodate to the variable need of parallelization
- out-of-use cores are waiting in low energy consumption mode
- hierarchic core-to-core communication greatly increases memory throughput
- asynchronous-style computing [32] largely reduces loss stemming from the gap [33] between speeds of processor and memory
- *principle of locality can be applied inside the processor*: direct core-to-core connection (more dynamic than in [34]) greatly enhances efficacy in large systems [35]
- the communication/computation ratio, defining decisively efficiency [11, 15, 36], is reduced considerably
- QTs thread-like feature akin to *fork()* and hierarchic buses change the dependence of the time of creating many threads on the number of cores from linear to logarithmic (enables to build exascale supercomputers)
- inter-core communication can be organized in some sense similar to Local Area Network (LAN)s of computer networking. For cooperating, cores can prefer cores in their topological proximity
- as the processor itself can candle scheduling signals in HW and in most cases the number of runnable tasks does not exceed the number of available computing resources, the conventional scheduling in multi-tasking systems can be reduced considerably

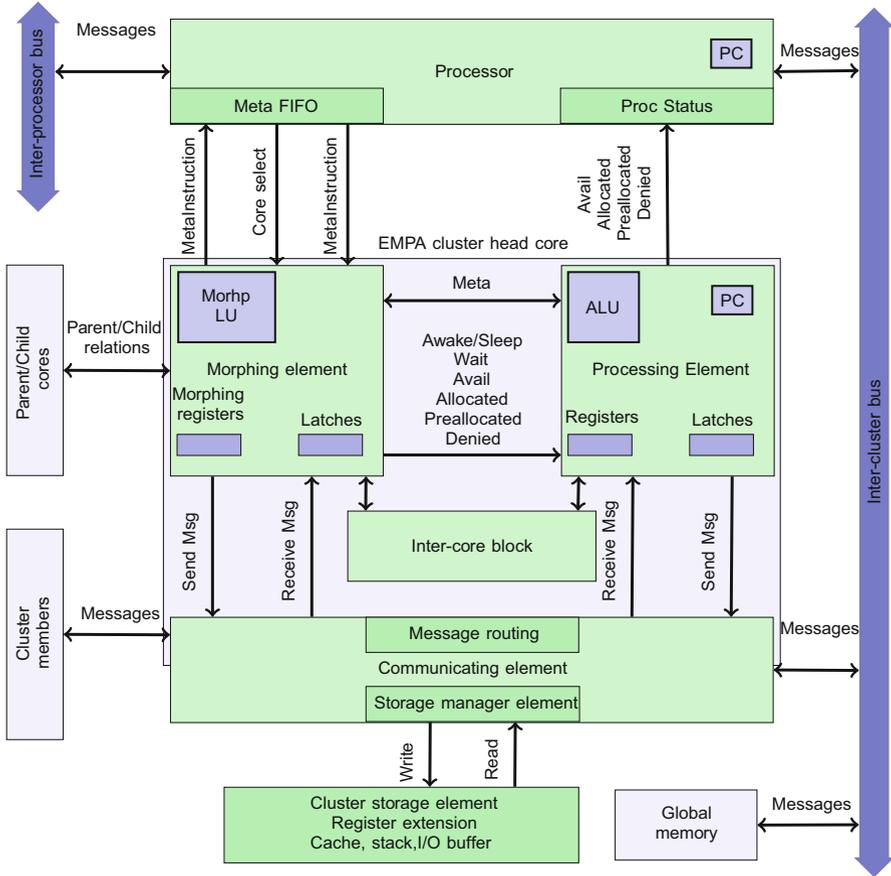


Fig. 2 The logical overview of the EMPA-based computing.

3 How to Implement EMPA

The best starting point to understand implementation of EMPA principles is conventional many-core processors. Present electronic technology made kilo-core processors available [37, 38], in a very inexpensive way and in immediate proximity of each other, in this way making the computing elements a “free resource” [39]. Principles of SPA, however, enable us to use them in a rather ineffective way [40]. Their temporal behavior [12] not only makes general-purpose cores ineffective [41], but their mode of utilization (mainly: their interconnection) leads to very low efficiency in performing real-life tasks [16, 42].

Given that true parallelism cannot be achieved (working with components anyhow needs time and synchronization via signals and/or messages, the question is only time resolution), *EMPA targets an enhanced and synchronized paral-*

lelized sequential processing based on using many cooperating processors. The implementation uses variable granularity and as much truly parallel portions as possible. However, *focus is on the optimization of the operation of the system, rather than providing some new kind of parallelization.* Ideas of cooperation comprise job outsourcing, sharing different resources and providing specialized many-core computing primitives in addition to single-processor instructions; as well as explicitly introducing different types of memory.

In this way *EMPA is an extension of SPA:* conventional computing is considered consisting of a single non-granulated thread, where (mostly) SW imitates the required illusion of granulating and synchronizing code fragments. Mainly because of this, many of components have a name and/or functionality familiar from conventional computing. Furthermore, we consider the *computing process as a whole* to be the subject of optimization rather than segregated components individually.

In SPA, there is only one active element, the Central Processing Unit (CPU). The rest of components of the system serves requests from CPU in a passive way. As EMPA wants to *extend* conventional computing, rather than to *replace* it, its operating principle is somewhat similar to the conventional one, with important differences in some key points. Figure 2 provides an overview of operating principle and major components of EMPA. We follow hints by Amdahl: “*general purpose computers with a generalized interconnection of memories or as specialized computers with geometrically related memory interconnections and controlled by one or more instruction streams*” [2].

3.1 The Core

An EMPA core of course comprises an EMPA Processing Element (EPE). Furthermore, it addresses two key deficiencies of conventional computing: inflexibility of computing architecture by EMPA Morphing Element (EME), and lack of autonomous communication by EMPA Communicating Element (ECE). Notice the important difference to conventional computing: *the next instruction can be taken either from memory pointed out by the instruction pointer (conventional instruction) or from the Meta FIFO (morphing instruction).*

The Processing Element

The EPE receives an address, fetches the instruction (if needed, also its operands). If the fetched instruction is a meta-instruction, EPE sets its ‘Meta’ signal (changes to ‘Morphing’ regime) for the EME and waits (suspends processing instructions) until the EME clears that signal.

The Morphing Element

When EPE sets ‘Meta’ signal, EME comes into play. Since the instruction and its operands are available, it attempts to process the received meta-instruction. However, the meta-instruction refers to resources handled by the processor. At processor level, order of execution of meta-instructions depends on their priority. Meta-instructions, however, may handle the ‘Wait’ of the core signal correspondingly. Notice that the idea is different from configurable spatial accelerator [43, 44]: the needed configuration is assembled ad-hoc, rather than chosen from a list of preconfigured assemblies.

Unlike in SPA, communication is a native feature of EMPA cores and it is implemented by ECE. Core assemble message content (including addresses), then after setting a signal, the message is routed to its destination, without involving a computing element and without any respect to where destination is. Message finds its path to its destination autonomously, using EMPA’s hierarchic bus system and ECEs of the fellow cores, taking the shortest (in terms of transfer time) path. Sending messages is transparent for both programmer and EPE.

The Storage Management Element

EMPA Storage Manager Element (ESME) is implemented only in cluster head cores, and its task is to manage storage-related messages passing through ECE. It has the functionality (among others) similar to that of memory management unit and cache controller in conventional computing.

3.2 Executing the Code

The Quasi-Threads

Code (here it means a reasonably sized sequence of instructions) execution begins with ‘hiring’ a core: the cores by default are in a ‘core pool’, in low energy consumption mode. The ‘hiring core’ asks for a helper core from its processor. If no cores are available at that moment, the processor sets the ‘Wait’ signal for the requester core and keeps its request pending. At a later time, processor can serve this pending request with a ‘reprocessed’ core.

Notice that the idea is quite different from the idea of eXplicit MultiThreading [45, 46]. Although they share some ideas such as the need for fine-grained multi-threaded programming model and architectural support for concurrently executing multiple contexts on-chip, unlike XMTs, QTs embody not simply mapping the idea of multi-threading to HW level. QTs are based on a completely unconventional computing paradigm; they can be nested.

This operating principle also means that code fragment and active core exist only together, and this combination (called Quasi-Thread) has a lifetime. Principle of the implementation is akin to that of the ‘dynamic variable’. EMPA hires a core for executing a well-defined code fragment, and only for the period between creating and terminating a QT. In two different executions, the same code fraction may run on different physical cores.

Process of Code Execution

When a new task fragment appears, an EMPA processor must provide a new computing resource for that task fragment (a new register file is available). Since an executing core is ‘hired’ only for the period of executing a specific code fragment, it must be returned to core pool when execution of the task fragment terminates. The ‘hired’ PU is working on behalf of the ‘hiring’ core, so it must have the essential information needed for performing the delegated task. Core-to-core register messages provide a way to transfer register contents from a parent core to a child core.

Beginning execution of an instruction sets signal ‘Meta’, i.e. selects either EPE or EME for the execution, and that element executes the requested action. The acting core repeats the process until it finds and ‘end of code fragment’ code. Notice the difference to conventional computing: processing of the task does not terminate; only the core is put back into ‘core pool’ as at the moment it is not anymore needed.

When ‘hired’ core becomes available, processing continues with fetching an instruction by the ‘hired’ core. For this, the core sends a message with the address of the location of the instruction. The requested memory content arrives at the core in a reply message logically from the addressed memory, but the ESME typically intercepts the action. The process is similar to the one in conventional computing. However, here memory triggers sending a reply to the request when it finds the requested contents, rather than keeping the bus busy. Different local memories, such as addressable cache, can also be handled. Notice also that the system uses complete messages (rather than simple signals with the address); this makes possible accessing some content independently from its location, although it needs location-dependent time.

Of course, ‘hiring’ core wants to get back some results from the ‘hired’ core. When starting a new QT, ‘hiring’ core also defines, with sending a mask, which register contents the hired core shall send back. In this case, synchronization is a serious issue: parent core utilizes its registers for its task, so it is not allowed to overwrite any of its registers without an explicit request from parent. Because of this, when a child terminates, it writes the expected register contents to a latch storage of the parent, then it may go back to ‘core pool’. When parent core reaches the point where it needs register contents received from its child, it explicitly asks to clone the required contents from latches to its corresponding register(s). It is the parent’s responsibility to issue this command at such a time when no accidental register overwriting can take place.

Notice that beginning execution of a new code fragment needs more resources, while terminating it frees some resources. Because of this, terminating a QT has a higher priority than creating one. This policy, combined with that cores are able to wait until their processor can provide the requested amount of resources, prevents “eating up” computing resources when the task (comprising virtually an infinite number of QTs) execution begins.

Compatibility with Conventional Computing

Conventional code shall run on an EMPA processor (as an implicitly created QT). However, that code can only use a single core, since it contains no meta-instructions to create more QTs. This feature enables us to mix EMPA-aware code with conventional code, and (among others) to use the plethora of standard libraries without rewriting that code.

Synchronizing the Cooperation

The cores execute their instruction sequences independently, but their operation must be synchronized at several points. Their initial synchronization is trivial: processing begins when the ‘hired’ core received all its required operands (including instruction pointer, core state, initial register contents, mask of registers the contents of which the hiring core requests to return). The final synchronization on the side of ‘hired’ core is simple: the core simply sends contents of the registers as was requested at the beginning of executing the code fragment.

On the side of a ‘hiring’ core, the case is much more complex. The ‘hiring’ core may wait for the termination of the code fragment running on the ‘hired’ core, or maybe it is in the middle of its processing. In the former case, a simple waiting until the message arrives is sufficient, but in the latter case, receiving some new register contents in some inopportune time would destroy its processing. Because of this, register contents from the ‘hired’ core are copied to the corresponding registers only when the ‘hiring’ core requests so explicitly. Figure 3 attempts to illustrate the complex cooperation between EMPA components.

3.3 Organizing ‘ad hoc’ Structures

EMPA can ‘morph’ internal architecture of the EMPA processor, as required by the actual task (fragment). EMPA uses principle of creating ‘parent-child’ (rather than ‘Master-Slave’) relation between its cores. The ‘hiring’ core becomes parent, and the ‘hired’ core becomes child. A child has only one parent, but parents can have any number of children. Children can become parents in some next phase of execution; in this way, several ‘generations’ can cooperate. This principle provides

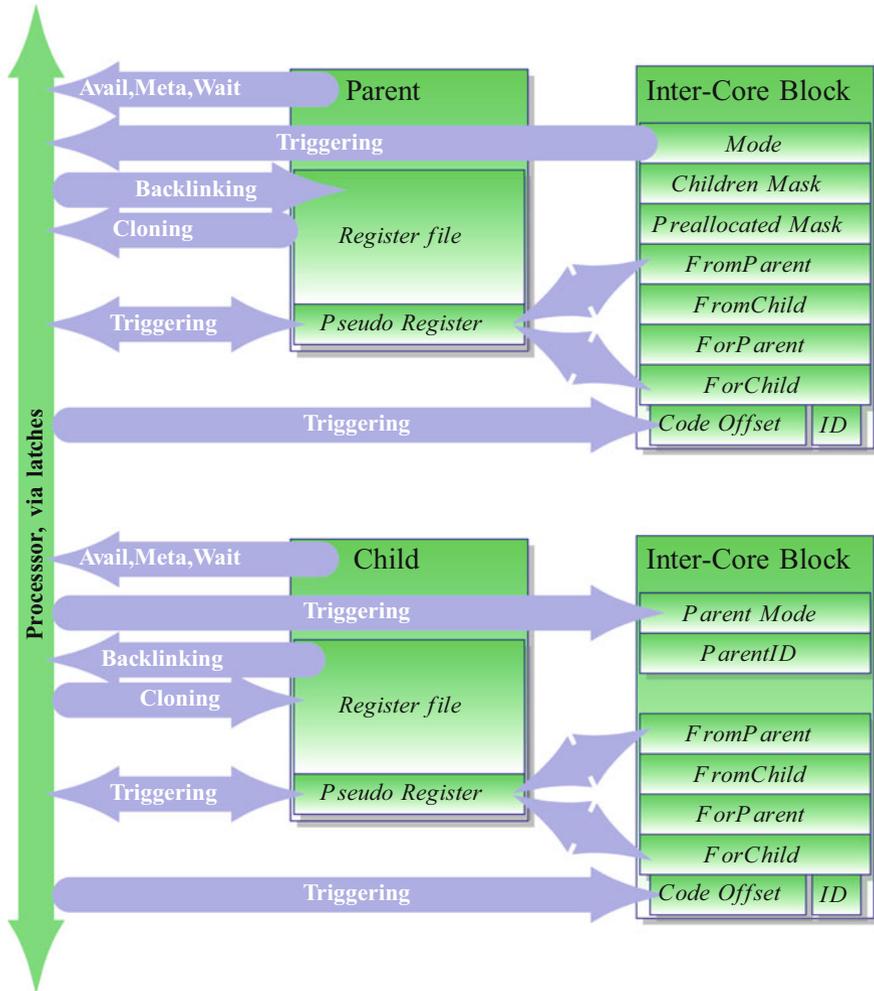


Fig. 3 Implementing the parent-child relationships: registers and operations of the EICB

a dynamic processing capacity for different tasks (in different phases of execution). The ‘parent-child’ relations simply mean storing addressing information, in the case of children combined with concluding the address from ‘hot’ bits of a mask.

As ‘parents are responsible for their children’, parents cannot terminate their code execution until all their children returned result of the code fragment that their parent delegated for them. This method enables parents also to trust in their children: when they delegate some fragment of their code to their children, they can assume that that code fragment is (logically) executed. It is the task of compiler

to provide the required dependence information, how those code fragments can be synchronized.

This fundamental cooperation method enables the purest form of delegating code to existing (and available) cores. In this way, all available processing capacity can be used, while only the actually used cores need energy supply (and dissipate). *Despite its simplicity, this feature enables us to make subroutine calls without needing to save/restore contents through memory and to implement mutexes working thousands of times quicker than in conventional computing.*

3.4 Processor

Processor comprises many physical EMPA cores. An EMPA processor appears in role of a ‘manager’ rather than a number-crunching unit, it only manages its resources.

Although individual cores initiate meta-instructions, their synchronized operation requires the assistance of their processor. Meta-instructions received by EMPA cores are written first (without authorization) in a priority-ordered queue (Meta FIFO) in the processor, so the processor can always read and execute only the highest priority meta-instruction (a core can have at most one active meta-instruction).

3.5 Clustering the Cores

The idea of arranging EMPA cores to form clusters is somewhat similar to that of CNNs [47]. In computing technology, one of the most severe limitations is defined by internal wiring, both for internal signal propagation time and area occupied on the chip [1]. In conventional architectures, cores are physically arranged to form a 2-dimensional rectangular grid matrix. Because of SPA, there should not be any connection between segregated cores, so the inter-core area is only used by some kind of internal interconnection networks or another wiring.

In EMPA processors, even-numbered columns in the grid are shifted up by a half grid position. In this way cores are arranged in a way that they have common boundaries with cores in their neighboring columns. In addition to these neighboring cores, cores have (up to two) neighbors in their column, with altogether up to six immediate neighbors, with common boundaries. This method of positioning also means that cores, logically, can be arranged to form a hexagonal grid, as shown in Fig. 4. Cores *physically* have a rectangular shape with joint boundaries with their neighbors, but *logically* they form a hexagonal grid. This positioning enables to form “clusters” of cores, forming a “flower”: an orange *ovary* (the cluster head) and six *petals* (the leaf cores of cluster, the members).

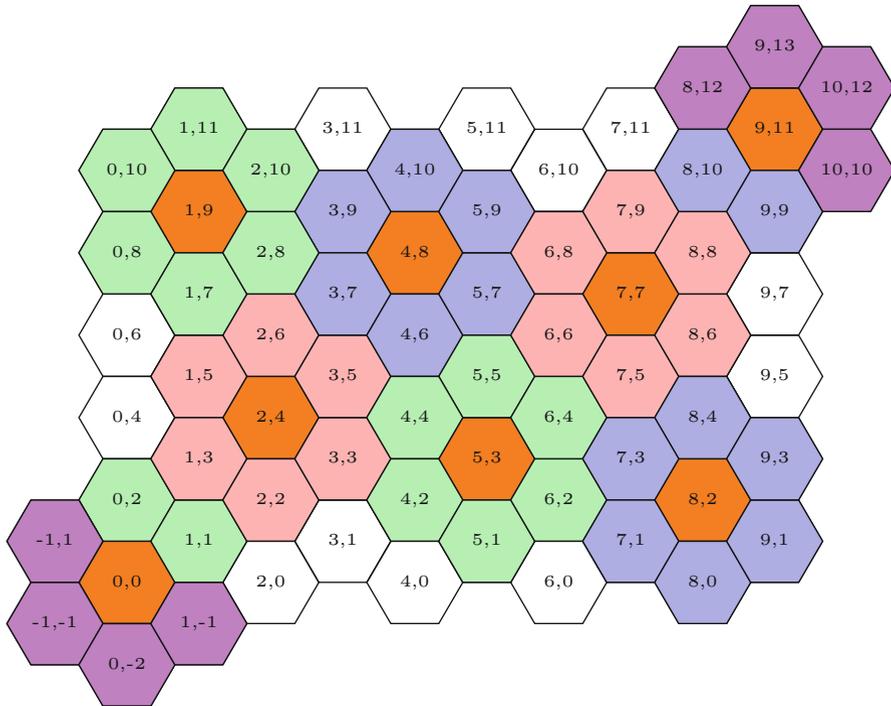
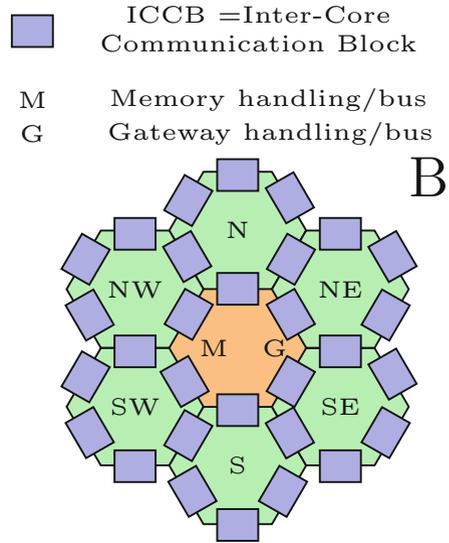


Fig. 4 The (logically) hexagonal arrangement (internal clustering) of EMPA cores in the EMPA processor

Between cores arranged in this way also neighborhood size can be interpreted similarly to the case of cellular computing. Based on neighborhood of size $r = 1$ (that means that cores have precisely one common boundary), a cluster comprising up to six cores (cluster members) can be formed, with the orange cell (of size $r = 0$, the cluster head) in the middle. Cluster members have shared boundaries with their immediate neighbors, including their cluster head. These cores define the external boundary of the cluster (the “flower”). Cores within this external boundary are “ordinary members” of the cluster, and the one in the central position is *head of the cluster*.

There are also “corresponding members” (of size $r = 2$): cores having at least one common boundary with one of the “ordinary members” of the cluster. “Corresponding members” may or may not have their cluster head, but have a common boundary with one of the “ordinary members”. White cells in the figure represent “external members” (also of size $r = 2$): they have at least one common boundary with an “ordinary member”, like the “corresponding members”, but unlike the “corresponding members” they do not have their cluster head. Also, there are some “phantom members” (see the violet petals in the figure) around the square edges in the figure: they have a cluster head and the corresponding cluster address,

Fig. 5 Implementation of the zeroth-level communication bus in EMPA



but (as they are physically not implemented in the square grid of cores during the manufacturing process) they do not exist physically.

That means: a cluster has one core as “cluster head”; up to six “ordinary members”, and up to twelve “corresponding members”; i.e., an “extended cluster” can also be formed, comprising up to 1+6+12 members. Notice that around the edge of the square grid “external members” can be in the position of the “corresponding members”, but the upper limit of the total number of members in an extended cluster does not change. Interpreting members of size $r \geq 2$ has no practical importance. *The cores with $r \leq 2$ have a direct communication mechanism (Fig. 5).*

3.6 Communication in EMPA

As discussed in [14], communication strongly degrades computing performance, even in relatively small-size computing systems [15, 21]. Its basic reason is the shared medium (whether it is physically Ethernet-like or serial connection), so EMPA introduces network-like addressing scheme, organizes traffic and introduces hierarchic bus system, to implement principle of locality at HW level.

Addressing and transport systems must provide support for all transport modes. Cluster addressing is of central importance because of the topology of cores: *cores having common boundary surely do not need a bus between the neighboring cores.* In this sense, the native, cross-boundary data transfer represents a zeroth-level communication bus (actually several, parallelly working “buses”), with no contention. This feature, combined with the “small world” nature of most computing tasks (especially the biology mimicking ones) and that nearby cores can share

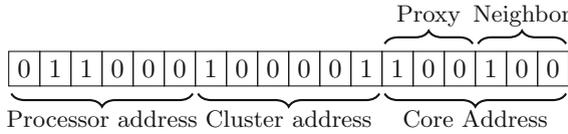


Fig. 6 Implementing the hierarchical cluster-based addressing bit fields of the cores of EMPA processors. A cluster address is globally unique.

memory contents available through the cluster head core) results in a serious performance boost. These architectural changes shall be useful for future neuromorphic architectures/applications, as AIs represent a very specific workload.³

Addressing (see also Fig. 6) must support the goal to keep messages at the lowest level of communication buses. Messages from/to outside the cluster are received/sent by cluster head. The rest of the messages are sent directly (through the corresponding Inter-Core Communication Block (ICCB)) or with using a proxy to their final destination. To implement that goal, EMPA processors use the addressing scheme shown in Fig. 6. Notice that the proposed addressing system a network logical address can be directly (and transparently) mapped to the ID and vice versa.

In EMPA, cluster addressing carries also topological information, partly relies on relative topological positions, and enables to introduce different classes of relationship between cells. As mentioned, cluster head cores have a physically distinct role (In this sense, they can also be a “fat” core) and enables us to introduce cluster addressing for members of the extended clusters. Only cluster head cores have an immediate global memory access, see Fig. 5 (considerably reducing the need for wiring). The cores being in neighborhood of size $r = 1$ can access memory through their cluster head. These cores can also be used as a proxy for cores in neighborhood of size $r = 2$. The latter feature also enables to replace a denied cluster head core.

In SPA, the grid and linear addressing are purely logical ones, which use absolute addresses known at compile time. Similarly to computer networks, EMPA cores have (closely related through the cluster architecture) both logical and physical addresses, enabling autonomous (computing-unrelated) communication and virtual addressing.

3.7 The Compiler

Compiler plays a significant role in EMPA. It should discover all possibilities of cooperation, especially the ones that become newly available with the philosophy

³<https://www.nextplatform.com/2019/10/30/cray-revamps-clusterstor-for-the-exascale-era/>: *artificial intelligence, ... it's the most disruptive workload from an I/O pattern perspective.*

that *appearance of a new task is attached with appearance of new computing resource, with a new register file*. Because at the time of compilation actual HW availability cannot be known, code for different scenarios must be prepared and put in the object code.

The philosophy of coding must be drastically changed. Given that, with outsourcing, a new computing facility appears, and processor assures proper synchronization and data transfer, there is no need to store/restore return address and save/restore data in the registers, leading to less memory traffic, and quicker execution time.

Object code is essentially unchanged, except that some fragments (the QTs) are bracketed by meta-instructions. The QTs can be nested (i.e., meta-instructions are inserted into conventional code). One can consider that QTs represent a kind of atomic macros which have some input and output register contents but do not need processing capacity from the actual core.

4 New Features EMPA Offers

Although EMPA does not want to address *all* challenges of computing, it addresses many of them (and leaves the door open for addressing further challenges). Due to lack of space, code examples, comparisons, and evaluations, based on the loosely-timed SystemC simulation [48], are left for simulator documentation and the early published version [49].

4.1 Architectural Aspects

Notice that ad hoc assemblies consider both current state of cores, and also their ‘Denied’ signal. That is, the flawed (or just temporarily overheated) cores are not used, significantly increasing mean time between machine failures. Also, notice that this approach enables using ‘hot swap’ cores, in this way providing dynamic, connected systems (the addressing is universal, and the information is delivered by messages; it takes time, but possible), as well as *to deliver the code to the data*: the physical cores can be located in the proximity of the ‘big data’ storage, instruction is delivered to the place, and only processed, needed result is to be transported back.

Virtualization at HW Level

In EMPA no absolute processor addresses are utilized: virtual processors seen by the programmer are mapped ‘on the fly’ to physical core by the EMPA processor. Physical cores have a ‘denied’ state that can be set permanently (like fabrication yield) or temporarily (like overheating), in which case the core will not be used to map a virtual core to it. When combined with a proper self-diagnostic system, this

feature prevents extensive systems to fail because of a failing core. Processor has the right and possibility to replace a physical core with another one at any time.

Redundancy

Huge masses (literally millions/billions) of silicon-based elements are deployed in all systems. As a consequence, components showing a tolerable error rate in “normal” systems, but (purely due to the high number of components) need special care in the case of large-scale systems [50].

The usual engineering practice is to rely on the high reliability of components. Fault-tolerant systems require particular technologies, typically majority voting, but they are also based on the same type of single high-reliability components.

Reduced Power Consumption

The operating principle of a processor is based on the assumption that processors are working continuously, executing instructions one after the other, as their control unit defines the required sequencing. Because of this principle, in the OS an ‘idle’ task is needed. In EMPA, cores can return control voluntarily, enabling most of the cores to stay in a ‘low power’ state.

Also, as discussed in [12], a major contribution of power consumption comes from moving data unnecessarily. Given that EMPA reduces memory usage in many ways (and, that according to [51], about 80% of consumed energy is used for moving data), it shall have a significant effect also of power efficiency of computing.

4.2 *Attacking Memory Wall*

The ‘memory wall’ is known as the ‘von Neumann’ bottleneck of computing, especially after that memory access time became hundreds of times slower than processing time. Although in SPA systems ‘register only’ processing and cache memories can seriously mitigate its effect, in the case of large systems the ‘sparse’ calculations that poorly use the cache, show up orders of magnitude worse computing efficacy, i.e., further improvement in using the memory is of utmost importance.

Register-to-Register Transfer

The idea of immediate register-to-register transfer [34] seriously can increase performance of real-life tasks [35]. In EMPA, the idea is used in combination with the flexibility of using virtual cores, multiple register arrays via children.

Subroutine Call Without Stack

In SPA, a subroutine call requires to save/restore return address and (at least part of) register file; unfortunately, one can use only main memory for that temporary storage. In EMPA, for executing subroutine code, another PU is provided. Because of this solution, HW can remember (in a nested way) the return address. Furthermore, working area is provided by the register file of the ‘hired’ core. Given that a register-to-register transfer is provided, code execution can be hundreds of times quicker. With proper organization, hiring and hired cores can also run partly parallel.

Interrupt and Systems Calls Without Context Switching

Given that interrupts and OS service calls can be considered as special service calls, where also context switching is needed, using a prepared (waiting in kernel mode) core can service a request thousands of times quicker. Event, interrupts can be serviced *without interrupting* the running process.

Resource Sharing Without Scheduling

For multitasking, only the OS can provide exclusive access to some resource (as in SPA, no other processor/task exists). EMPA offers a simple, elegant, and quick solution: it can delegate a QT for the task of guarding a critical section, and all tasks issue a conditional subroutine call to the code guarded by that QT. All but the first requester QT must wait (but are scheduled automatically by the processor), and after servicing all requests, the delegated core is put back to the pool. Since compiler creates reasonably sized code fragments, cases leading to priority inversion [30] cannot happen, so no specialized protocols are needed in the OS: the orchestrated work in EMPA prevents those issues.

4.3 Attacking the Communication Wall

In SPA, communication is not natively present (no other processor exists); it must be performed and synchronized using I/O instructions and OS operations, in payload processing time; resulting in performing a severe amount of non-payload instructions.

Decreasing the Internal Latency

When using interconnected cores, ECE can take over most of the non-payload duties, enabling to decrease the sequential-only portions of the task that decisively define communication/computation ratio [36]; a significant point when developing large scale computing systems [11] or using AI-type workloads [15]. As discussed in [11], the housekeeping (the FP_0) contribution is a considerable limitation when running High Performance Linpack (HPL) benchmark. Borrowing nearby cores and using their physical proximity enables us to achieve higher *HPL* maximum performance values.

Hierarchic (Local) Communication

Using temporally or spatially local memory accesses can increase efficiency dozens of times. Similarly, providing ‘interconnection cache’ for EMPA processor can result in considerable improvement in final efficiency of the system. As computing tasks change their state between ‘computing bound’ and ‘communication bound’ dynamically, this solution mitigates both limiting factors as much as possible.

Fully Asynchronous Operation

As von Neumann only required a ‘proper sequencing’ of instructions, and having less ‘idle’ times during core operation appears as performance increase, asynchronous operation (i.e., turning all components to active) can considerably contribute to more effective (i.e., comprising fewer losses) operation.

5 Summary

In computing, incremental development methods face more and more difficulties, because of the drastic changes both in technology and utilization. The final reason, as has been suspected by many researchers, is the computing paradigm reflecting a 70-year old state of the art. Computing needs renewal [49] and rebooting. Firstly, the ever smaller components driven by ever quicker clock signal, because of scientific reasons, show a temporal behavior [12], and suppressing their natural behavior causes severe computing performance loss (and enormously increased power consumption). Secondly, many technical implementations and architectural solutions, inherited from the past decades, become outdated. It was presented that *it is not a necessary condition that the same computer solves all the tasks*: von Neumann only required a “proper sequencing” in executing machine instructions. This requirement can be satisfied in a much better way via using the presently available many “free” processors. That way requires an entirely different thinking

(and component base) and offers real advantages. We can implement the introduced new paradigm by putting the presently available technology solutions along with different principles that approach offers considerable advantages.

References

1. I. Markov, Limits on fundamental limits to computation. *Nature* **512**(7513), 147–154 (2014)
2. G.M. Amdahl, Validity of the single processor approach to achieving large-scale computing capabilities, in *AFIPS Conference Proceedings*, vol. 30, pp. 483–485 (1967)
3. K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, K. Yelick, A view of the parallel computing landscape. *Commun. ACM* **52**(10), 56–67 (2009)
4. J.A. Chandy, J. Singaraju, Hardware parallelism vs. software parallelism, in *Proceedings of the First USENIX Conference on Hot Topics in Parallelism*, ser. HotPar '09 (USENIX Association, Berkeley, CA, USA, 2009), pp. 2-2
5. S.H. Fuller, L.I. Millett, Computing performance: Game over or next level? *Computer* **44**, 31–38 (2011)
6. US National Research Council, The Future of Computing Performance: Game Over or Next Level? (2011). [Online]. Available: <http://science.energy.gov//media/ascr/ascasc/pdf/meetings/mar11/Yelick.pdf>
7. S(o)OS Project, Resource-independent execution support on exa-scale systems (2010). <http://www.soos-project.eu/index.php/related-initiatives>
8. Machine Intelligence Research Institute, Erik DeBenedictis on supercomputing (2014). [Online]. Available: <https://intelligence.org/2014/04/03/erik-debenedictis/>
9. J. Sawada et al., TrueNorth ecosystem for brain-inspired computing: Scalable systems, software, and applications, in *SC '16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 130–141 (2016)
10. J. Végh, A. Tisan, The need for modern computing paradigm: Science applied to computing, in *Computational Science and Computational Intelligence CSCI The 25th Int'l Conf on Parallel and Distributed Processing Techniques and Applications* (IEEE, 2019), pp. 1523–1532. [Online]. Available: <http://arxiv.org/abs/1908.02651>
11. J. Végh, Finally, how many efficiencies the supercomputers have? *J. Supercomput.* **76**(12), 9430–9455 (2020). [Online]. Available: <http://link.springer.com/article/10.1007/s11227-020-03210-4>
12. J. Végh, Introducing temporal behavior to computing science, in *2020 CSCE, Fundamentals of Computing Science* (IEEE, 2020). Accepted FCS2930, in print. [Online]. Available: <https://arxiv.org/abs/2006.01128>
13. J. Végh, A.J. Berki, Do we know the operating principles of our computers better than those of our brain? (2020). [Online]. Available: <https://arxiv.org/abs/2005.05061>
14. J. Végh, Which scaling rule applies to Artificial Neural Networks, in *Computational Intelligence (CSCE) The 22nd Int'l Conf on Artificial Intelligence (ICAI'20)* (IEEE, 2020). Accepted ICA2246, in print; in review in *Neurocomputing*. [Online]. Available: <http://arxiv.org/abs/2005.08942>
15. J. Végh, How deep machine learning can be, ser. *A Closer Look at Convolutional Neural Networks* (Nova, In press, 2020), pp. 141–169. [Online]. Available: <https://arxiv.org/abs/2005.00872>
16. J. Végh, How Amdahl's Law limits performance of large artificial neural networks. *Brain Informatics* **6**, 1–11 (2019). [Online]. Available: <https://braininformatics.springeropen.com/articles/10.1186/s40708-019-0097-2/metrics>

17. J. Gubbi, R. Buyya, S. Marusic, M. Palaniswami, Internet of Things (IoT): A vision, architectural elements, and future directions. *Future Gener. Comput. Syst.* **29**, 1645–1660 (2013)
18. R.F. Service, Design for U.S. exascale computer takes shape. *Science* **359**, 617–618 (2018)
19. J. Du, L. Zhao, J. Feng, X. Chu, Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. *IEEE Trans. Commun.* **66**, 1594–1608 (2018)
20. www.top500.org, Intel dumps knights hill, future of xeon phi product line uncertain (2017). <https://www.top500.org/news/intel-dumps-knights-hillfuture-of-xeon-phi-product-line-uncertain//>
21. J. Keuper, F.-J. Preundt, Distributed training of deep neural networks: theoretical and practical limits of parallel scalability, in *2nd Workshop on Machine Learning in HPC Environments (MLHPC)* (IEEE, 2016), pp. 1469–1476. [Online]. Available: <https://www.researchgate.net/publication/308457837>
22. ARM, big.LITTLE technology (2011). [Online]. Available: <https://developer.arm.com/technologies/big-little>
23. J. Congy, et al., Accelerating sequential applications on CMPs using core spilling. *Parallel Distribut. Syst.* **18**, 1094–1107 (2007)
24. Cypress, CY7C026A: 16K x 16 Dual-Port Static RAM (2015). <http://www.cypress.com/documentation/datasheets/cy7c026a-16k-x-16-dual-port-static-ram>
25. R. Banakar, S. Steinke, B.-S. Lee, M. Balakrishnan, P. Marwedel, Scratchpad memory: Design alternative for cache on-chip memory in embedded systems, in *Proceedings of the Tenth International Symposium on Hardware/Software Codesign*, ser. CODES '02 (ACM, New York, NY, USA, 2002), pp. 73–78. [Online]. Available: <http://doi.acm.org/10.1145/774789.774805>
26. J. Backus, Can programming languages Be liberated from the von Neumann style? A functional style and its algebra of programs. *Commun. ACM* **21**, 613–641 (1978)
27. P. Gohil, J. Horn, J. He, A. Papageorgiou, C. Poole, IBM CICS Asynchronous API: Concurrent Processing Made Simple (2017). <http://www.redbooks.ibm.com/redbooks/pdfs/sg248411.pdf>
28. R.H. Arpaci-Dusseau, A.C. Arpaci-Dusseau, *Operating Systems: Three Easy Pieces*, 0th edn. (Arpaci-Dusseau Books, 2015)
29. J. Végh, A new kind of parallelism and its programming in the explicitly many-processor approach. ArXiv e-prints (Aug. 2016). [Online]. Available: <http://adsabs.harvard.edu/abs/2016arXiv160807155V>
30. O. Babaoglu, K. Marzullo, F.B. Schneider, A formalization of priority inversion. *Real Time Syst.* **5**(4), 285–303 (1993). [Online]. Available: <https://doi.org/10.1007/BF01088832>
31. D.W. Wall, Limits of instruction-level parallelism, New York, NY, USA, pp. 176–188 (Apr. 1991). [Online]. Available: <http://doi.acm.org/10.1145/106974.106991>
32. S. Kumar, et al., Acceleration of an asynchronous message driven programming paradigm on ibm blue gene/q, in *2013 IEEE 27th International Symposium on Parallel and Distributed Processing* (IEEE, Boston, 2013). [Online]. Available: <https://ieeexplore.ieee.org/abstract/document/6569854>
33. N. Satish, C. Kim, J. Chhugani, H. Saito, R. Krishnaiyer, M. Smelyanskiy, M. Girkar, P. Dubey, Can traditional programming bridge the ninja performance gap for parallel computing applications? *Commun. ACM* **58**(5), 77–86 (2015). [Online]. Available: <http://doi.acm.org/10.1145/2742910>
34. F. Zheng, H.-L. Li, H. Lv, F. Guo, X.-H. Xu, X.-H. Xie, Cooperative computing techniques for a deeply fused and heterogeneous many-core processor architecture. *J. Comput. Sci. Technol.* **30**(1), 145–162 (2015)
35. Y. Ao, C. Yang, F. Liu, W. Yin, L. Jiang, Q. Sun, Performance optimization of the HPCG benchmark on the sunway TaihuLight dupercomputer. *ACM Trans. Archit. Code Optim.* **15**(1), 11:1–11:20 (2018)
36. J.P. Singh, J.L. Hennessy, A. Gupta, Scaling parallel programs for multiprocessors: Methodology and examples. *Computer* **26**(7), 42–50 (1993)

37. B. Bohnenstiehl, A. Stillmaker, J.J. Pimentel, T. Andreas, B. Liu, A.T. Tran, E. Adeagbo, B.M. Baas, KiloCore: A 32-nm 1000-processor computational array. *IEEE J. Solid State Circuits* **52**(4), 891–902 (2017)
38. PEZY, 2048 core chip (2017). <https://www.top500.org/green500/lists/2017/11/>
39. S.B. Furber, D.R. Lester, L.A. Plana, J.D. Garside, E. Painkras, S. Temple, A.D. Brown, Overview of the SpiNNaker system architecture. *IEEE Trans. Comput.* **62**(12), 2454–2467 (2013)
40. M.D. Hill, M.R. Marty, Amdahl’s law in the multicore era. *IEEE Computer* **41**(7), 33–38 (2008)
41. R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B.C. Lee, S. Richardson, C. Kozyrakis, M. Horowitz, Understanding sources of inefficiency in general-purpose chips, in *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ser. ISCA ’10 (ACM, New York, NY, USA, 2010), pp. 37–47. [Online]. Available: <http://doi.acm.org/10.1145/1815961.1815968>
42. J. Végh, J. Vásárhelyi, D. Drótos, The performance wall of large parallel computing systems, in *Lecture Notes in Networks and Systems*, vol. 68 (Springer, 2019), pp. 224–237. [Online]. Available: <https://link.springer.com/chapter/10.1007%2F978-3-030-12450-221>
43. K.E. Fleming Jr., K.D. Glossop, S.C. Steely Jr., J. Tang, A.G. Gara, Processors, methods, and systems with a configurable spatial accelerator, no. 20180189231 (July 2018). [Online]. Available: <http://www.freepatentsonline.com/y2018/0189231.html>
44. Intel, Processors, methods and systems with a configurable spatial accelerator (2018). <http://www.freepatentsonline.com/y2018/0189231.html>
45. U. Vishkin, Explicit multi-threading (XMT): A PRAM-on-chip vision – A desktop supercomputer (2007). Last accessed Dec. 12, 2015 [Online]. <http://www.umiacs.umd.edu/users/vishkin/XMT/index.shtml>
46. U.Y. Vishkin, Spawn-join instruction set architecture for providing explicit multithreading (1998). <https://patents.google.com/patent/US6463527B1/en>
47. V. Cimagalli, M. Balsi, Cellular neural networks: A review, in *Proc. 6th Italian Workshop on Parallel Architectures and Neural Networks, Vietri sul Mare, Italy* (World Scientific, 1993), pp. 12–14. iSBN: 9789814534604
48. J. Végh, EMPAthY86: A cycle accurate simulator for explicitly many-processor approach (EMPA) computer (Jul 2016). [Online]. Available: <https://github.com/jvegh/EMPAthY86>
49. J. Végh, *Renewing Computing Paradigms for More Efficient Parallelization of Single-Threads*, ser. *Advances in Parallel Computing*, vol. 29, ch. 13 (IOS Press, 2018), pp. 305–330. [Online]. Available: <https://arxiv.org/abs/1803.04784>
50. C. Wired, Cosmic Ray Showers Crash Supercomputers. Here’s What to Do About It (2018). <https://www.wired.com/story/cosmic-ray-showers-crashsupercomputers-heres-what-to-do-about-it/>
51. H. Simon, Why we need Exascale and why we won’t get there by 2020, in *Exascale Radioastronomy Meeting*, ser. AASCTS2, 2014. [Online]. Available: <https://www.researchgate.net/publication/261879110> Why we need Exascale and why we won’t get there by 2020

Formal Specification and Verification of Timing Behavior in Safety-Critical IoT Systems



Yangli Jia, Zhenling Zhang, Xinyu Cao, and Haitao Wang

1 Introduction

The Internet of things (IoT) emerges as a common platform and service for consumer electronics [1]. IoT systems can be deployed into safety-critical missions such as defense, traffic, process control, environmental control, automotive systems, medical service, etc., and a failure in the temporal aspect in these systems can be as critical as one in the functional aspect and many, directly affecting the environment and lives of people [2]. To efficiently guarantee the high quality of these safety-critical IoT systems, it is necessary to clearly model, visualize, and verify the systems' interaction behavior before deploying them.

Formal modeling methods have the characteristics such as consistent, concise, unambiguous, and precise clarity, and we can also visualize and verify the behaviors based on the formal specification. Therefore it has great significance to improve accuracy, reliability, security of the systems by formal modeling, visualization, and verification of the behavior of complex IoT systems [3].

Many methods have been presented for modeling systems' interaction and other behavior properties [4]. These methods can be divided into two different categories. The first set of behavior specification methods can be called automata theory-based methods, such as duration automata [5], timed automata [6], timed I/O automata [7], and timed interface automata [8]. Timed automata has clock variable to express timing constraint information, while duration automata gives timing

Y. Jia · Z. Zhang (✉)

School of Computer Science & Technology, Liaocheng University, Liaocheng, China
e-mail: jiayangli@lcu.edu.cn; zhangzhenling@lcu.edu.cn

X. Cao · H. Wang

China National Institute of Standardization, Beijing, China
e-mail: caoxy@cnis.ac.cn; wanght@cnis.ac.cn

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_32

459

constraints by binding a simple upper bound and lower bound for each transition. The other set of component behavior specification methods is usually based on process algebras, such as timed CSP [9], hybrid CSP [10], and behavior protocol [11]. All of these methods are easy to learn and usually supported by automated verification, but as without considering structural aspect of components, such behavior modeling techniques focus only on the behavioral aspect and are unable to describe the interconnection structure of hierarchical component architecture which also influences the behavior. This is one of the reasons why these models are often considered unavailable and cannot describe the time constraint information in IoT systems [12].

In this paper, we use formal method to model the interactive actions and give a visualization of these interaction behaviors. The formal model can be used easily to specify and verify IoT systems' interaction behavior and timing constraint information.

The rest of this paper is organized as follows. Section 2 focuses on the specification language of enhanced time behavior protocol. Composition and verification of timing behavior are discussed in Sect. 3. Section 4 gives an application example. The last section concludes the paper with the future work.

2 Specification Language for Modeling of IoT Systems' Behavior

2.1 Behavior Protocol

Behavior protocol [11] is a formalism used to describe abstract model of software systems by a set of admissible sequences of method calls. In behavior protocol, every method call or a return from a method call forms an event which can be denoted by event tokens. For a method named m , event tokens $!m^{\wedge}$, $?m^{\wedge}$, $!m^{\#}$, and $?m^{\#}$ stand for emitting a method call, accepting a method call, emitting a return, and accepting a return. We can use the basic operators (eg. “;”, “+”, and “*”), the enhanced operators (eg. “|”, “||”, and “/”), and the composed operators (eg. “ \cap_x ”, “|T|”) to construct a behavior protocol in a way similar to a regular expression. A sequence of event tokens denoting events occurs in a component of the system form a trace. Thus, the trace $\langle !m^{\wedge}; ?m^{\#} \rangle$ describes the activity of a caller (emitting a method call followed by accepting the return), while the trace $\langle ?m^{\wedge}; !m^{\#} \rangle$ denotes what the call does (accepting a call and emitting the return).

Behavior protocol uses regular-like expressions to specify systems' interaction behavior and provides clear support for behavior specification and formal reasoning about the correctness of behavior. In addition, it is easy to read and apply.

2.2 Enhanced Time Behavior Protocol

We extend behavior protocol to the area of IoT systems and the result is enhanced time behavior protocol (ETBP). ETBP programs model systems’ interaction behavior as a sequence of timed communication events and timed internal events. Events in ETBP are bound with a timing constraint like duration automata and a 2-tuple time consumption constraint according to requirements of IoT applications. ETBP programs are easy to read and apply just as BP programs. Based on the advantages of both behavior protocol and duration automata, ETBP can be used easily to specify and verify IoT systems’ interaction behavior and timing constraint information. The syntax and semantics of enhanced time behavior protocol owe much to timed CSP [9].

We can specify the time delay conditions and the time consumption constraint by means of timed event tokens. Inspired by duration automata [5], we extend event tokens by binding timing constraint and time consumption information on each event token needed. So timed event tokens take the forms of $!m[t_1][t_2,t_3]^{\wedge}$, $?m[t_1][t_2,t_3]^{\wedge}$, $!m[t_1][t_2,t_3]^{\#}$, and $?m[t_1][t_2,t_3]^{\#}$.

For example, $!interface.method[t_1][t_2,t_3]^{\wedge}$ stands for that invoking component of IoT system emit a method call request within t_1 time units delay and the event’s time consumption interval is $[t_2,t_3]$, while $?interface.method[t_1][t_2,t_3]^{\wedge}$ means the request must be accepted by the target component of IoT system within t_1 time units. $!interface.method[t_1][t_2,t_3]^{\#}$ stands for the target component emits a response delayed for at most t_3 time units, and $?interface.method[t_1][t_2,t_3]^{\#}$ means the invoking component accepts the response within t_4 time units. All the invoke and acceptance events must be done in time interval $[t_2,t_3]$.

The time consumption constraint is represented as a 2-tuple $[a,b]$, where $a,b \in \mathbb{N}$ and $a \leq b$. The relative time consumption interval $[a,b]$ is a time consumption interval measured with respect to some reference time consumption instant. We represent it by a natural number. The number starts from 0 and increases by 1 each time when a new tick is generated.

Time consumption constraints will be used as guards for enhanced time behavior protocol.

Basic operators of ETBP are defined in classical regular expressions.

Enhanced operators in ETBP include the parallel operators “|” and “||”, the restriction operator “/”, the timeout operator, the delay operator “Idle,” the guard operator “when,” and the reset operator and the “Stop” operator.

If we use A, B denotes a protocol. Timed BP can be defined by the following Backus-Naur form:

$$P ::= A; B \mid A + B \mid A^* \mid A \triangleright_t B \mid A \mid B \mid A \mid B \mid A/G \mid \text{Reset}(t) \mid \text{Idle}(t) \\ \text{When} \mid \text{Stop} \mid A \cap_x B.$$

Basic operators in ETBP include the sequencing operator “;”, the alternative operator “+”, and the repetition operation “*”.

- $A;B$ represents a succession of protocol A by protocol B. The set of event traces formed by $A;B$ are a concatenation of a trace generated by A and a trace generated by B.
- $A+B$ represents an alternative of protocol A and protocol B. The set of event traces formed by $A+B$ are generated either by A or by B.
- A^* represents an repetition of protocol A., and it is equivalent to $NULL + A + (A;A) + (A;A;A) + \dots$ where A is repeated any finite number of times.

Then we give the enhanced operators including the parallel operators “|” and “||” and the restriction operator “/”.

- $A|B$ represents the “and-parallel” of protocol A and protocol B. The set of event traces formed by $A|B$ are an arbitrary interleaving of event tokens of traces generated by A and B.
- $A||B$ represents the “or-parallel” of protocol A and protocol B. It stands for $A + B + (A|B)$.
- A/G represents the “restriction” of protocol A. The event tokens not in a set G are omitted from the traces generated by A.

Besides operators in behavior protocol, we give other time-related operators to ETBP to construct more complicated protocols. These operators include the timeout operator, the delay operator “Idle,” the guard operator “when,” the reset operator, and the “Stop” operator.

- $A \triangleright_t B$ expresses the timeout operation of protocol A and protocol B. It may execute any events that A may perform before time t, but if a timeout occurs, it will execute events that B may perform.
- When operator takes the form (when b A) which means that it will perform events trace generated by A if b is true, otherwise the events trace generated by A cannot be chosen for execution.
- Idle(t) is the delay operator. It does nothing but waits t time units.
- Reset(t) operator just reset the clock variable t to 0.

Stop can be used to specify a component behavior which does nothing except terminate.

2.3 Example of Enhanced Time Behavior Protocol

Consider the protocol $?a;(!p[2][8, 9]+!q[2][8, 9])\triangleright_2!b[2][8, 9]||?c$. It contains event tokens $?a, !p[2][8, 9], !q[2][8, 9], !b[2][8, 9], ?c$ and the operators $;, +, \triangleright_2,$ and $||$. (In the examples here, we omit event suffixes “” and “#” and connection names for simplicity.) The protocol generates traces including, for instance, $\langle ?a; !p[2][8, 9] \triangleright_2 !b[2][8, 9] \rangle, \langle ?a; !q[2][8, 9] \triangleright_2 !b[2][8, 9] \rangle, \langle ?a; ?c; !p[2][8, 9] \triangleright_2 !b[2][8, 9] \rangle,$ and $\langle ?a; !p[2][8, 9] \triangleright_2 !b[2][8, 9]; ?c \rangle$. The trace $\langle ?a; !p[2][8, 9] \triangleright_2 !b[2][8, 9] \rangle$ starts with $?a$, followed by $!p[2][8, 9] \triangleright_2 !b[2][8, 9]$, which means emitting the

method call p within 2 time units delay, and if timeout occurs, it will emit the method call b within 2 time units. All the invoke and acceptance events must be done in time interval [8, 9]

3 Composition and Visualization of ETBP

To better support for complex real-time component-based systems' development, we must provide compatibility verification theory and automation tools for the ETBP model.

3.1 Composition of Enhanced Time Behavior Protocol

For behavior protocol P and behavior protocol Q , we use $P \cap_x Q$ to give the definition of composition of enhanced time behavior protocol P and Q . Their composition must meet the timing constraint information of the application. In the case of composition, the time behavior protocol P or Q sends an event, the time behavior protocol Q or P receives an event, the time behavior protocols P and Q interact with the event, and internal events are generated.

If their composition meets the timing constraint information, any appearance of $!interface.method[t][t_2, t_3]$, $?interface.method[t][t_2, t_3]$, resp. $?interface.method[t][t_2, t_3]$, and $!interface.method[t][t_2, t_3]$, as a result of the interleaving, is merged into $\tau interface.method[t]$ in the resulting trace, and the invoke and acceptance events must be done in time interval $[t_2, t_3]$ if every event has a time consumption constraint.

In an ideal case, the event perfectly matches, that is, the time behavior protocol P or Q executes the "send an event," and the corresponding behavior protocol Q or P exactly executes the "receive an event," the time consumption constraint also satisfied, and then the two time behavior protocols P and Q interact with the event to generate internal events.

For example: the enhanced time behavior protocols $P = !tm.begin[t_1][t_3, t_4]; (!tm.commit[t_1][t_3, t_4] + !tm.rollback[t_1][t_3, t_4])$ and $Q = ?tm.begin[t_1][t_3, t_4]; ?tm.rollback[t_1][t_3, t_4]$, $P \cap_x Q$ create the path $\tau tm.begin; \tau tm.rollback$. The combined enhanced time behavior protocol does not contain events $tm.commit$, because the right side of the combination operator requires that events $rollback$ must occur and events $rollback$ and $commit$ can only be executed by one of them.

In order to describe composite operations accurately, we give the formal operational semantics of composite operations, and the interaction between P and Q needs to be considered. Suppose A and B are event sets of time behavior protocol P and Q , respectively. In the case of interaction, protocol P or Q executes the sending operation, and the protocol Q or P executes the receiving event. P and Q interact

with the event on this way, and internal events are generated in the case of combined operation.

From an evolutionary perspective,

$$\frac{P \xrightarrow{t} P', Q \xrightarrow{t} Q'}{P \cap_X Q \xrightarrow{t} P' \cap_X Q'}$$

From a migration perspective,

$$\frac{P \xrightarrow{(t, !a)} P', Q \xrightarrow{(t, ?a)} Q'}{P \cap_X Q \xrightarrow{(t, !a \wedge ?a)} P' \cap_X Q'} \quad [!a \in A, ?a \in B, a \in X]$$

3.2 Composition and Verification of Behavior Protocols

We give the composition and verification algorithm of timing behaviors as follows. Algorithm input: enhanced time behavior protocol P, Q, event set X, Algorithm output: the composition result of P, Q. (1) read the two protocol P, Q, classified the events in P, Q into the sets $S_{P, prov}$, $S_{P, req}$, $S_{Q, prov}$, $S_{Q, req}$ (2) while not end of protocol P or Q, (3) if there is a call event $?m^{\wedge}[t][t_2, t_3]$ or $!m^{\#}[t][t_2, t_3]$ in P or Q, (4) then traverse Q or P to find if there is any event m in the form of $?m^{\wedge}[t][t_2, t_3]$ or $?m^{\#}[t][t_2, t_3]$, generates the composition traces $tc = !m^{\wedge}[t][t_2, t_3]?m^{\wedge}[t][t_2, t_3]$, or $tc = !m^{\#}[t][t_2, t_3]?m^{\#}[t][t_2, t_3]$, $m \in X$ (5) $T(c) = Utc_i$ (6) traverse all T (c), (7) any $tc \in T(c)$ (8) while tc is not terminated (9) if there is trace $tc1 = !m^{\wedge}[t_1][t_3, t_4]?m^{\wedge}[t_2][t_3, t_4]$, or, $tc2 = !m^{\#}[t_1][t_3, t_4]?m^{\#}[t_2][t_3, t_4]$, $m \in X$ (10) then (11) if there is overlap of $[t_1]$, $[t_2]$ and done in $[t_3, t_4]$ (12) then combined as $\tau m[t]$ (13) else output the path, and invalid timed activity error (14) if there is $m \in (S_{P, prov} \cup S_{Q, prov}) \wedge m \in X$ (15) then output the path, and invalid timed activity error (16) if there is $m \in (S_{P, req} \cup S_{Q, req}) \wedge m \in X$ (17) then output the path, and stop forward error (18) $tc \leftarrow tc'$, $tc' \in T(c)$, goto (8)

Obviously, the algorithm gives the composition result, and compatibility of two protocol is verified by determining the related compatible errors in composition. Based on the composition algorithm, we can visualize the process and the result.

Based on the LTSA tool [13] which has an extensible architecture allowing extra features to be added by means of plugins, we have developed an integrated tool named ETBPSV for specifying and verifying IoT systems' behavior.

4 Application of ETBP

To demonstrate the specification model described above, consider the case of a boiler pressure control real-time system. The boiler pressure control real-time system consists of four components including pressure sensor, pressure monitor, pressure controller, and alert. The pressure monitor component acquires pressure data from the pressure sensor component every 18 ms and sends information to the pressure controller component within 2 ms to request the issue of cooling or warming if the pressure is too high or too low. The controller responds to the information within 1 ms. If error occurs, the system will give an alarm and stop running within 1 ms.

We give the formal specification of enhanced time behavior protocol for the components in the system as follows:

```

Pressure_sensor:
?pressure^;!pressure#;
Pressure_monitor:
(!pressure^;?pressure#;
(
!low^ [2]▷2 (!alert^;?alert#;Stop) [1];?low#; Idle(18-t)
+!high^ [2]▷2 (!alert^;?alert#;Stop) [1];?high#; Idle(18-t)
+Idle(18);
)
)*;
where t is the delay time for the pressure monitors sending
      request information of cooling or warming to the
      controller, and t<2.
Pressure_controller:
?low^;!low#[1]+?high^;!high#[1];
Alert:
(?alert^;!alert#) or ?alert.
The four sub-components can be combined into a
      composition-component (system/subsystem), which behavior as:
(τpressure^;τpressure#;
(
τlow^ [2]▷2 ( τalert^;τalert#;Stop) [1];τlow#; Idle(18-t)
+τhigh^ [2]▷2 ( τalert^;τalert#;Stop) [1];τhigh#; Idle(18-t)
+Idle(18);
)
)*;

```

By using the time-related operators in ETBP, components' real-time behavior can be formally specified easily and precisely in designing component-based IoT systems. Obviously, combining the advantages of both simplicity and practicality, ETBP has more powerful description ability and can be used easily to specify real-time components' behavior and timing constraint information. The behavior protocols of real-time sub-components can be composed together to build high-level protocols based on the composition definition. And based on the formal specification, we can analyze and verify the timeliness, safety, and other trustworthiness properties of component-based IoT systems.

We input the enhanced time behavior protocol of the three components into ETBPSV. After compiling, we can see the states migration diagram of each protocol as shown in Figs. 1, 2, and 3, respectively.

The behavior protocol composition migration diagram of pressure sensor and pressure monitor is shown in Fig. 4.

As the composite protocol will send out event $!tem_low^2$, when it combines with time behavior protocol of pressure controller, the corresponding response event



Fig. 1 States migration diagram of pressure sensor's protocol

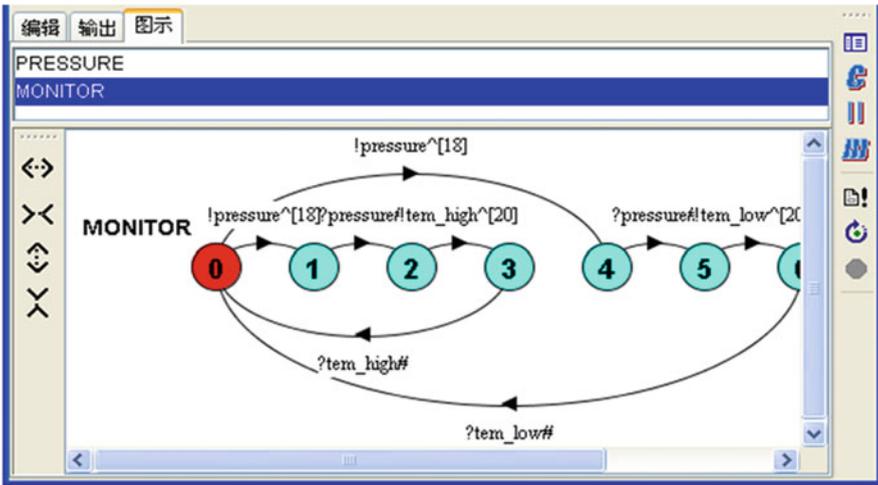


Fig. 2 States migration diagram of pressure monitor's protocol

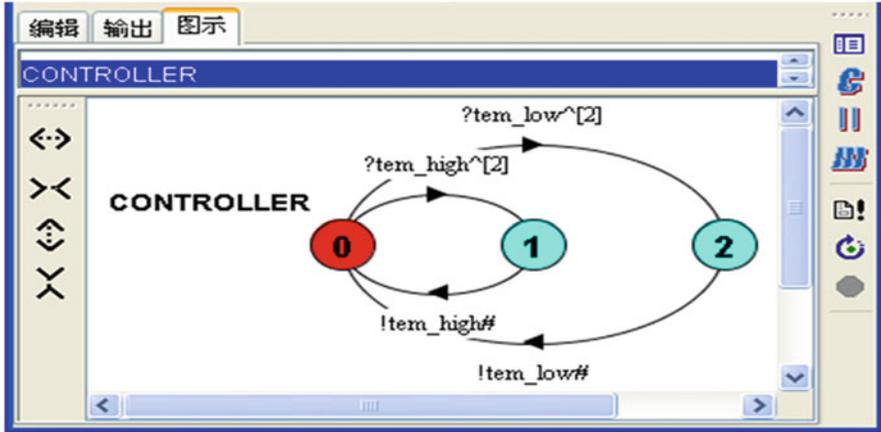


Fig. 3 States migration diagram of pressure controller's protocol

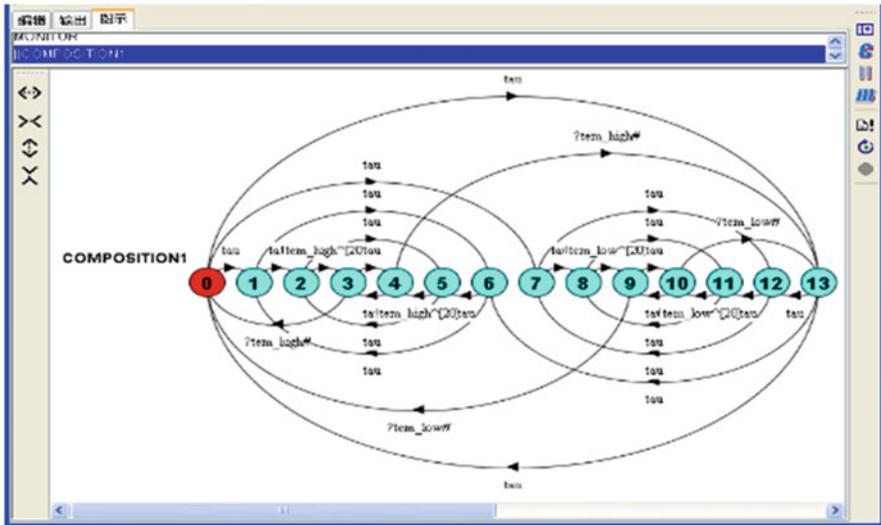


Fig. 4 States migration diagram of the combined two protocols

?tem_high^2 cannot be found; the combined result will give bad activity prompt, as shown in Fig. 5.

Three behavior protocols can be combined into one, and the graphical results can be displayed using ETBPSV as Fig. 6. It can be seen that the diagram is very complex. Therefore, it is unrealistic to analyze the results of time behavior protocol combination manually. We can formally verify in ETBPSV as shown in Fig. 7.

We have experimentally specified several specifications of IoT systems modeled by enhanced time behavior protocol, and these specifications were visualized

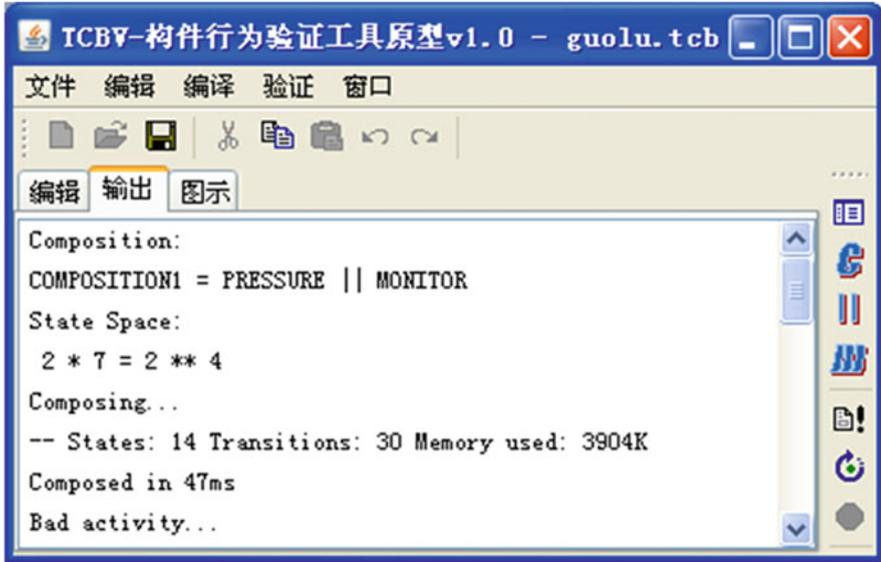


Fig. 5 Verification of the combined two protocols

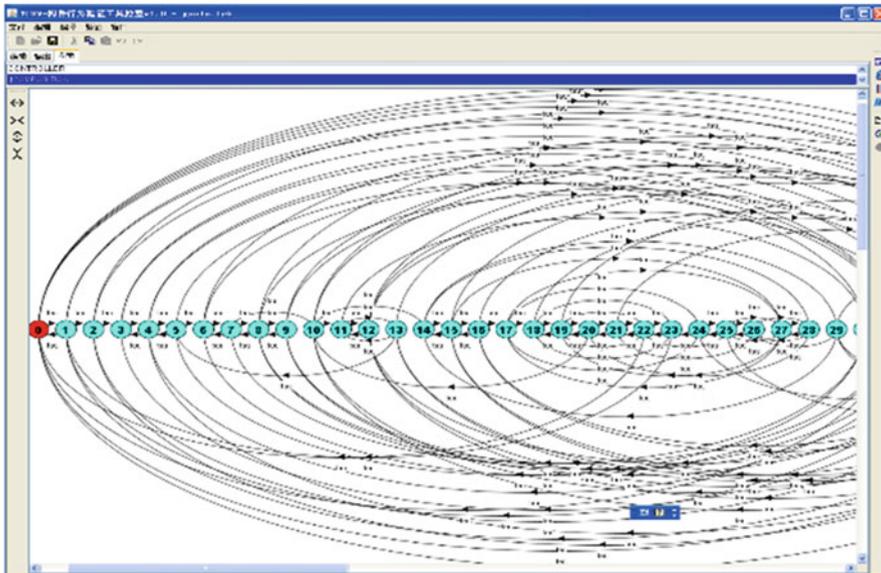


Fig. 6 Migration diagram of the combined three protocols

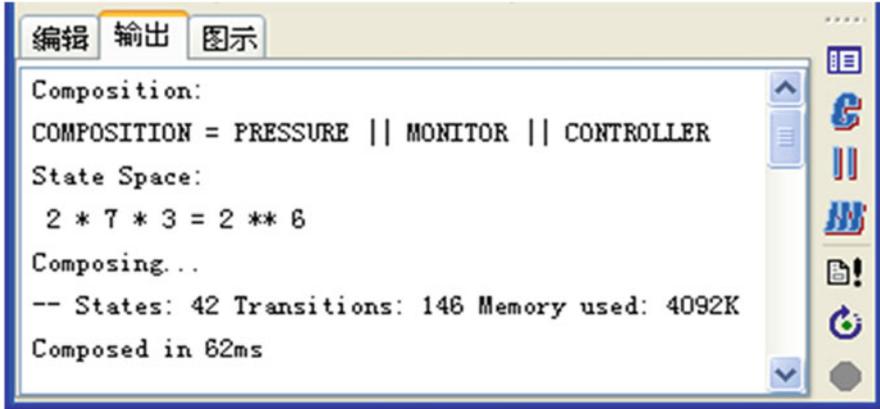


Fig. 7 Verification result of the composition

and verified using ETBPSV, and some errors about timeliness, safety, and other trustworthiness properties were found effectively. But with the complex increasing of ETBP model, state space explosion problem is becoming significantly aggravated and affect the efficiency of visualization and verification seriously.

5 Conclusions and Future Work

We presented a formal specification method of timing interaction behavior in IoT systems. In addition, we developed a more efficient automatic verification framework based on enhanced time behavior protocol and gave an example to show how the method can be used. Combining the advantages of both simplicity and practicality, the enhanced time behavior protocol has more powerful description ability and can be used easily to specify real-time interaction behavior and timing constraint information which provide a rich base for further application of formal methods.

As future work related to timing behavior specification and verification, we intend to focus on (1) giving the semantics of operators in enhanced time behavior protocol and (2) dedicating to the research of state space reduction algorithm.

Acknowledgments This work is supported by the National Natural Science Foundation of China under Grant No. 81973695 and Soft Scientific Research Project of Shandong Province under Grant No. 2018RKB01080.

References

1. A. Čolaković, M. Hadžialić, Internet of Things (IoT): A review of enabling technologies, challenges, and open research issues. *Comput. Netw.* **144**, 17–39 (2018)
2. K. Hofer-Schmitz, B. Stojanović, Towards formal verification of IoT protocols: A review. *Comput. Netw.*, 107233. ISSN 1389-1286. (2020). <https://doi.org/10.1016/j.comnet.2020.107233>
3. S. Tang, D.R. Shelden, C.M. Eastman, et al., A review of building information modeling (BIM) and the internet of things (IoT) devices integration: Present status and future trends. *Autom. Constr.* **101**(May), 127–139 (2019)
4. P. Fremantle, B. Aziz, Deriving event data sharing in IoT systems using formal modelling and analysis, *Internet of Things*. **8**, 100092, ISSN 2542-6605 (2019). <https://doi.org/10.1016/j.iot.2019.100092>
5. D.V. Hung, B.V. Anh, Model checking real-time component based systems with blackbox testing, in *Proceedings of IEEE RTCSA'05*, Washington, DC, 2005, pp. 76–79
6. F. Heidarian, J. Schmaltz, F.W. Vaandrager, Analysis of a clock synchronization protocol for wireless sensor networks. *Theor. Comput. Sci.* **413**(1), 87–105 (2012)
7. D.K. Kaynar, N. Lynch, R. Segala, F. Vaandrager, *The Theory of Timed I/O Automata[R]* (MIT Laboratory for Computer Science, Cambridge MA, 2004)
8. L. de Alfaro, T.A. Henzinger, M. Stoelinga, Timed interfaces[C], in *Proceedings of the Second International Workshop on Embedded Software*, (Springer, Berlin, 2002), pp. 108–122
9. J. Davies, S. Schneider, A brief history of timed csp. *Theor. Comput. Sci.* **138**(2), 243–271 (1995)
10. H. Jifeng, From CSP to hybrid systems, in “*A Classical Mind, Essays in Honour of C.A.R. Hoare*, International Series in Computer Science, ed. by A. W. Roscoe, (Prentice Hall, 1994), pp. 171–189
11. F. Plasil, S. Visnovsky, Behavior protocols for software components. *IEEE Trans. Softw. Eng.* **28**(11), 1056–1076 (2002)
12. L. Brim, I. Cerna, P. Varekova, B. Zimmerova, Component-interaction automata as a verification-oriented component-based system specification. *SIGSOFT Software Engineering Notes* **31**(2) (2006)
13. J. Magee, J. Kramer, *Concurrency-State Models and Java Programs* (Wiley, 1999)

Introducing Temporal Behavior to Computing Science



János Végh 

1 Introduction

Computing science is on the border of mathematics and, through its physical implementation, science. Since the beginning of computing, the computing paradigm itself, “*the implicit hardware/software contract [1]*”, defined how mathematics-based theory and its science-based implementation must cooperate. Mathematics, however, considers only the *dependencies* between its operands; it assumes that the needed operands are instantly available. That is, computing science considers that performing operations, delivering operands to and from processing units, is as kind of engineering imperfectness. At the time when von Neumann proposed his famous abstraction, both time of processing and time of accessing data (including those on a mass storage device) were in the milliseconds region, while physical data delivery time was in the range of microseconds, i.e., three orders of magnitude smaller. *It was a plausible assumption to consider that total time of processing comprises only time of computation plus time of data access; data delivery time was neglected.*

For today, however, technical development changed the relations between those timings drastically. Today the data access time is much larger than the time needed to process them. Besides, the relative weight of the data transfer time has grown tremendously, for many reasons. Firstly, miniaturizing the processors to sub-millimeter size, while keeping the rest of the components (such as buses) above the centimeter scale. Secondly, the single-processor performance stalled [2],

Project no. 136496 has been implemented with the support provided from the National Research, Development and Innovation Fund of Hungary, financed under the K funding scheme.

J. Végh (✉)
Kalimános BT, Debrecen, Hungary

mainly because of reaching the limits, the laws of nature enable [3]. Thirdly, making truly parallel computers failed [1], and to reach the needed high computing performance we need to put together an excessive number of segregated processors. This latter way replaces *parallel computing* with *parallelized sequential computing*, disregarding that the operating rules of that different kind of computing [4–6] sharply differ from those of the segregated processors. Fourthly, the mode of utilization (mainly multitasking) forced out using operating system (OS), which imitates a “new processor” for a new task, at serious time expenses. Finally, the idea of “real-time connected everything” introduced geographically large distances with the corresponding several millisecond data delivery times. Theory of computing kept the idea of “instant delivery”; although even within the core, wiring has an increasing role. The idea of non-temporal behavior was confirmed by accepting “weak scaling” [7], suggesting that *all housekeeping times, such as organizing joint work of parallelized serial processors, sharing resources, using exceptions and OS services, delivering data between processing units and data storage units, are negligible*.

Vast computing systems can cope with their tasks with growing difficulty, enormously decreasing computing efficiency, and enormously growing energy consumption; one can experience similar issues in the world of networked edge devices. Being not aware of that collaboration between processors needs a different approach (another paradigm), resulted in demonstrative failures already known (such as supercomputers Gyoukou and Aurora’18, or brain simulator SpiNNaker)¹ and many more may follow: such as Aurora’21 [9], the China mystic supercomputers² and the EU planned supercomputers.³ General-purpose computing systems comprising “only” millions of processors already show the issues, and brain-like systems want to comprise four orders of magnitude higher number of computing elements. When targeting neuromorphic features such as “deep learning training”, the issues start to manifest already at a couple of dozens of processors [10, 11]. The scaling is nonlinear [5], strongly depending on the workload type, and the Artificial Intelligence (AI)-class workload is one of the worst workloads [5, 11] one can run on conventional architectures.⁴

“Successfully addressing these challenges [of neuromorphic computing] will lead to a new class of computers and systems architectures” [12]. However, the roundtable concentrated *only* on finding new materials and different gate devices. *They did not even mention that for such systems new computing paradigm may*

¹The explanations are quite different: Gyoukou was withdrawn after its first appearance; Aurora failed: retargeted and delayed; Despite the failure of SpiNNaker1, the SpiNNaker2 is also under construction [8]; “Chinese decision-makers decided to withhold the country’s newest Shuguang supercomputers even though they operate more than 50 percent faster than the best current US machines”.

²<https://www.scmp.com/tech/policy/article/3015997/china-has-decided-not-fan-flames-super-computing-rivalry-amid-us>.

³https://ec.europa.eu/newsroom/dae/document.cfm?doc_id=60156.

⁴<https://www.nextplatform.com/2019/10/30/cray-revamps-clusterstor-for-the-exascale-era/>: artificial intelligence, ... it’s the most disruptive workload from an I/O pattern perspective.

also be needed. The result was that, as noticed by judges of the Gordon Bell Prize, “surprisingly, [among the winners of the supercomputer competition] there have been no brain-inspired massively parallel specialized computers” [13]. Despite the vast need and investments, furthermore the concentrated and coordinated efforts, just because of the vital bottleneck: *the missing theory*.

2 Introducing Time to Computing

As suspected by many experts, the computing paradigm itself, “*the implicit hardware/software contract* [1]”, is responsible for the experienced issues: “*No current programming model is able to cope with this development [of processors], though, as they essentially still follow the classical van Neumann model*” [14]. When thinking about “advances beyond 2020”, the solution was expected from the “*more efficient implementation of the von Neumann architecture*” [15], however.

There are many analogies between science and computing [16]; among others, how they handle time. Both classic science and classic computing assume instant (infinitely quick) interaction between its objects. That is, an event happening at any location can be instantly seen at all other locations: time has no specific role, and an event has immediate effect on all other considered objects. In science, discovering that the speed of light is insurmountable, led to introducing the *four-dimensional space-time*. Special relativity introduces a ‘fourth space dimension’, and *we calculate that coordinate of the Minkowski space from the time as the distance the light traverses in a given time*.

2.1 Why Temporal Logic Is Needed

In computing, distances get defined during fabrication of components and assembling the system. In biological systems, nature defines neuronal distances, and in ‘wet’ neuro-biology, signal timing rather than axon length is the right (measurable) parameter. To describe temporal operation of computing systems correctly, *we need to find out how much later a component notices that an event occurred in the system*. To introduce a *temporal logic* (i.e. that value of a logical expression depends on where and when it is evaluated) into computing, the *reverse* of Minkowski transform is required: we need to use a special 4-vector, where all coordinates are time values: the first three are the corresponding local coordinates (distances from the location of the event, divided by the speed of interaction) having time dimension, and the fourth coordinate is the time itself; that is, we introduce a *4 dimensional time-space* system. The resemblance with the Minkowski-space is obvious, and the name difference signals the different aspects of utilization.

Figure 1a shows *why time must be considered explicitly in all kinds of computing*. The figure shows (for visibility) a 3-dimensional coordinate system: how an event

behaves in a two-dimensional space plus time (the concept is easier to visualize with the number of spatial dimensions reduced from three to two). In the figure, the direction ‘y’ is not used, but enables to place observers at the same distance from the event, without the need to locate them in the same point. The event happens at point (0,0,0), the observers are located on the ‘x’ axis; the vertical scale corresponds to the time.

In the classic physical hypothetical experiment, we switch on a light in the origo, and the observer switches his light when notices that the first light was switched on. If we graph the growing circle with the vertical axis of the graph representing time, the result is a cone, known as the *future light cone* (in 2D space plus a time dimension). Both light sources have some “processing time”, that passes between noticing the light (receiving the instruction) and switching the light on (performing the instruction). That is, the instruction is received at the origo, at the bottom of the green arrow. The light goes on at the head of the arrow, (i.e., at the same location, but at a later time), after that the ‘processing time’ T_p passed. Following that, the light propagates in the two spatial dimensions as a circle around axis “t”. Observers at larger distance notice the light at a later time: a ‘transmission time’ T_t is needed. If “processing time” of the light source of the first event were zero, the light would propagate along the gray surface at the origo. However, because of the finite processing time, the light will propagate along the blueish cone surface, at the head of the green arrow.

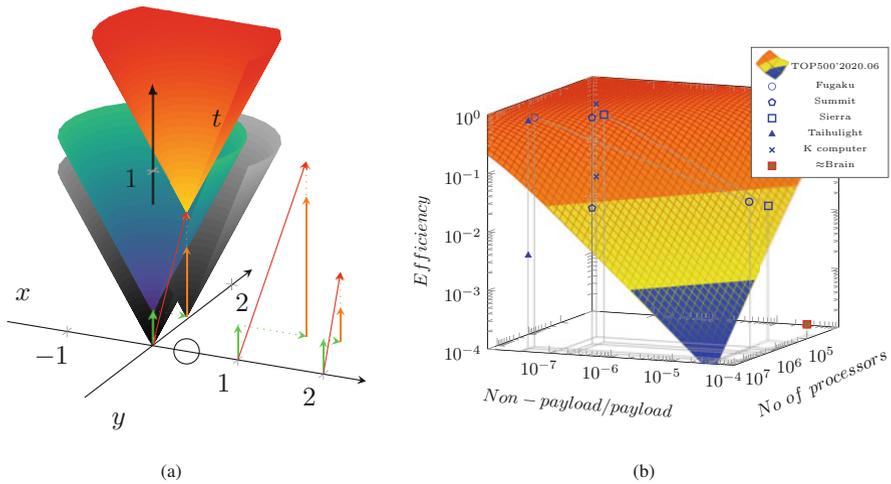


Fig. 1 The origin of “idle waiting time” and its effect on the efficiency on parallelized sequential processing systems. **(a)** The computing operation in time-space approach. The processing operators can be gates, processors, neurons or networked computers. **(b)** The surface and the figure marks show at what efficiency the top supercomputers run the ‘best workload’ benchmark HPL, and the ‘real-life load’ HPCG [6]. The right bottom part displays the expected efficiency [17] of running neuromorphic calculations on SPA computers

A circle denotes position of our observer on the axis “x”. With zero “transmission time”, the second gray conical surface (at the head of the green dotted arrow) would describe his light. However, its “processing time” can only begin when the observer notices the light at his position: when the dotted red arrow hits the blueish surface. At that point begins “processing time” of the second light source; the yellowish conical surface describes the second light propagation. The horizontal (green dotted) arrow describes the physical distance of the observer (as a time coordinate), the vertical (red dotted) arrow describes the time delay of the observer light. It comprises two components: the T_t transmission time to the observer and its T_p processing time. The light cone of the observer starts at $t = 2 * T_p + T_t$.

The red arrow represents the resulting *apparent processing time* T_A : the longer is the red vector; the slower is the system. As the vectors are in the same plane, $T_A = \sqrt{T_t^2 + (2 \cdot T_p + T_t)^2}$, that is $T_A = T_p \cdot \sqrt{R^2 + (2 + R)^2}$. This means, that *the apparent time is a non-linear function of both of its component times and their ratio R*. If more computing elements are involved, T_t denotes the longest transmission time. (Similar statement is valid if the T_p times are different) The effect is significant: if $R = 1$, the apparent execution time of performing the two computations is more than 3 times longer than the processing time. Two more observers are located on the axis ‘x’, at the same position. For visibility, their timings are displayed at points ‘1’ and ‘2’, respectively. Their results illustrate the influence of the transmission speed (and/or the ratio R). In their case *the transmission speed differs by a factor of two* compared to that displayed at point ‘0’; in this way three different $R = T_t/T_p$ ratios are displayed.

Notice that at half transmission speed (the horizontal green arrow is twice as long as that in the origo) the vector is considerably longer, while at double transmission speed, the decrease of the time is much less expressed.⁵ Given that the *apparent processing time* T_A defines the performance of the system, T_p and T_t must be concerted.

2.2 Consequences of Temporal Behaviour

Notice an important aspect: *the T_p transmission time is an ‘idle time’* (the orange arrow on the figure) for the observer: it is ready to run, takes power, but does no useful work. Due to their finite physical size and limited interaction speed (both neglected in the classic paradigm), *temporal operation of computing systems results inherently in an idle time of their processing units*,⁶ and—since it sensitively depends on many factors and conditions—*can be a significant contributor to non-payload portion of their processing time*. With other major contributors, originating

⁵Reference [6] discusses this phenomenon in details.

⁶It can be a crucial factor of inefficiency of general-purpose chips [18].

Listing 1. The essential lines of source code of the one-bit adder implemented in SystemC

```
//We are making a 1-bit addition
aANDb = a.read() & b.read();
aXORb = a.read() ^ b.read();
cinANDaXORb = cin.read() & aXORb;

//Calculate sum and carry out
sum = aXORb ^ cin.read();
cout = aANDb | cinANDaXORb;
```

from their technical implementation (see Sect. 3.2), these “idle waiting” times sharply decrease payload performance of the systems. Figure 1b depicts how efficiencies of recent supercomputers depend [6] on the number of single-threaded processors in the system and the parameter $(1 - \alpha)$, describing non-payload portion of the corresponding benchmark task. It is known since decades that “*this decay in performance is not a fault of the architecture, but is dictated by the limited parallelism*” [4]; in excessive systems of modern hardware (HW), *is also dictated by laws of nature* [16].

Using shorter operands (half precision rather than double precision) reduces T_A non-proportionally: the housekeeping costs (such as fetching, addressing) remain constant (although the amount of data movement and manipulation decreases). One expects a four-fold performance increase when using half-precision rather than double precision operands [19], and the consumed power consumption data underpin that expectation. However, the measured increase in computing performance was only three times higher: the apparent execution time T_A and the processing time T_p differ.

2.3 Example: Temporal Diagram of a 1-Bit Adder

Although for its end-users, the processor is the “atomic unit” of processing,⁷ principles of computing are valid also at “sub-atomic” level of gate operations. Describing the temporal operation at gate level is an excellent example, that *the line-by-line compiling (sequential programming, called also Neumann-style programming [20]), formally introduces only logical dependence, but through its technical implementation it implicitly and inherently introduces a temporal behavior, too.*

The one-bit adder is one of the simplest circuits used in computing. Its common implementation comprises 5 logic gates, 3 input signals and 2 output signals. Gates

⁷The reconfigurable computing, with its customized processors and non-processor-like processing units, does not change significantly the landscape.

are logically connected internally: they provide input and output for each other. The relevant fraction of the equivalent source code is shown in Listing 1.

Figures 2a and b show the timing diagram of a one-bit adder, implemented using common logic gates. The three input signals are aligned on axis y , the five logic gates are aligned on axis x . Gates are ready to operate as well as signals are ready to be processed (at the head of the blue arrows). The logic gates have the same operating time (the length of green vectors), their access time includes the needed multiplexing. Signals must reach their gate (dotted green arrows), that (after its operating time passes) produces its output signal, that starts immediately towards the next gate. Vertical green arrows denote gate processing (one can project the arrow to axis x to find out the ID of the gate), labelled with the name of the produced signal. There are “pointless” arrows in the figure. For example, signal $a \& b$ reaches the OR gate much earlier, than the signal to its other input. Depending on the operands of OR , it may or may not result in the final sum.

Notice, that considering physical distance and finite interaction speed, drastically changes the picture we have (based on “classic computing”), that the operating time of an adder is simply the sum of the corresponding “gate times”. For example, the very first AND and XOR operations could work in parallel (at the same time), but the difference in their physical distance the signals must travel, changes the times when they can operate with their signals. Also, compare the temporal behavior of the signal sum on the two figures. The only difference between subfigures is that the second XOR gate moved to another place.

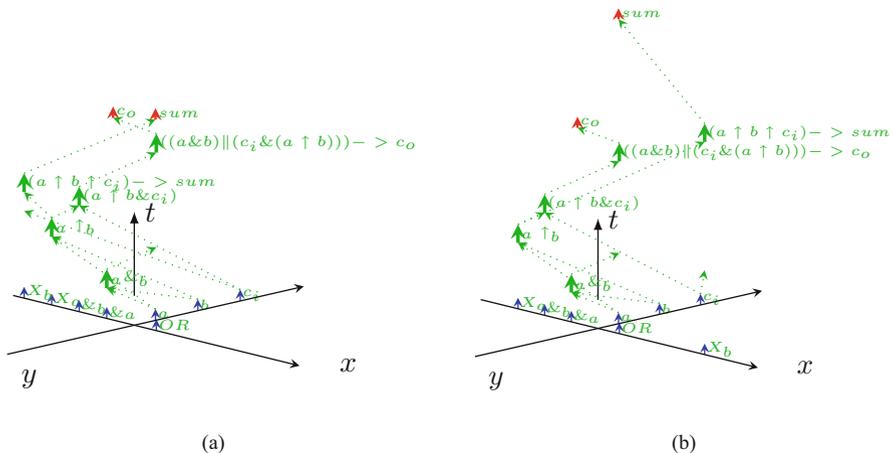


Fig. 2 Temporal diagram of a one-bit adder in time-space system. The diagram shows the logical equivalent of the SystemC source code of Listing 1., the time from axis x to the bottom of green arrows signals “idle waiting” time (undefined gate output). Notice how changing position of a gate affects signal timing. **(a)** Temporal dependence diagram of a 1-bit adder. The second XOR gate is at $(-1,0)$. **(b)** Temporal dependence diagram of a 1-bit adder. The second XOR gate is at $(+1,0)$.

The difference in timing roots not only in the different number of gates involved: the distance traversed by the signals can contribute equally, and even counterbalance the different number of involved gates. As the c_o output is the input c_i for the next bit, it must be wired there. The total execution time of, say, a 64-bit adder shall be optimized at that level, rather than at bit level. Orchestrating temporal operation through considering both complexity of operation, and positions of signals and operators, can significantly enhance performance.

The goal of this section and Figs. 2a and b is only to call the attention to that *in addition to the viewpoint of mathematics (using standard gates and logic functions) and technology (which technology enables to produce smaller gate times and smaller expenses), also the temporal behavior must be considered, when designing chips.* Even inside a simple adder circuit, the performance can be changed significantly, only via changing physical distance of gates; in strong contrast with the “classic computing”.

The meaning of “idle waiting” is slightly changed here, *The gates produce valid output only after they received all of their internally-produced operand(s), plus their “gate time”, at the head of the corresponding green arrow. The total operating time of the adder is considerably longer than the sum of operating times of its gates.* The proper positioning of gates (and wiring them) is a point to be considered seriously, and maybe also the role of gates must be rethought.

2.4 Using New Effect/Technology/Material in Computing Chain

Given that *apparent processing time* T_A defines performance of the system, T_p (physical processing time, a vector perpendicular to the XY plane) and T_t (transfer time, a vector between different planes) must be concerted. In a complex system, *it is not reasonable to fabricate smaller components without decreasing their processing time proportionally; and similarly, replacing a Processing Unit (PU) with a very much quicker one has only marginal effect, if the physical distance of the PUs cannot be reduced proportionally, at the same time.*

Figure 3 demonstrates why: two different topologies and two different physical cache operating speeds are used in the figure. Two cores are in positions $(-0.5,0)$ and $(0.5,0)$, furthermore two cache memories at $(0,0.5)$ and $(0,1)$. The signal, requesting to access cache, propagates along the dotted green vector (it changes both its time and position coordinates), the cache starts to operate only when the green dotted arrow hits its position. After its operating time (the vertical orange arrow), the result is delivered back to the requesting core. This time can also be projected back to the “position axes”, and their sum (red thin arrow) can be calculated. The physical delivery of the fetched value begins at the bottom of the lower vertical green arrows, includes waiting and finishes at the head of the upper vertical green arrows; their distance defines the *apparent cache access time* T_A . Physical cache

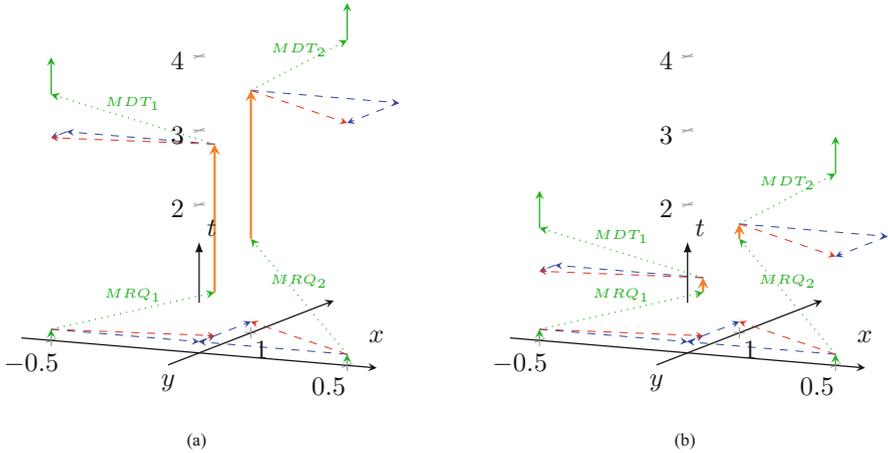


Fig. 3 Performance dependence of an on-chip cache memory, at different cache operating times, in the same topology. Cores at $x = -0.5$ and $x = 0.5$ positions access on-chip cache at $y = 0.5$ and $y = 1.0$, respectively. Vertical orange arrows represent physical cache operating time, and vertical green arrows the apparent access time. The physical operations speed of cache memory of the right subfigure is 10 times better. Compare the apparent access times to the corresponding physical ones (the time ratio is better only about a factor of two). Notice also that the apparent operating speed is more sensitive to the position rather than to the speed of the cache memory. **(a)** Normal speed cache memory. Two different cache memories, with the same physical cache speed, but at different internal on-chip cache position. **(b)** Super-quick (10 times quicker) cache memory. Assumes new material/physical mechanism. Two different cache memories, with the same physical cache speed, but at different internal on-chip cache position

access time (the vertical orange arrow) begins when signal reaches the cache. Till that time, cache is idle waiting. Core is also idle waiting until the requested content arrives. Notice that *apparent processing time is a monotonic function of the physical processing time, but because of the included—fixed time—‘transmission times’ due to physical distance of the respective elements, their dependence is far from being linear.* Repeated operation of course can change the idle/to active ratio; one must consider, however, the resources the signal delivery uses.

The apparent processing time (represented by the distance of the vertical green arrows) is only slightly affected by the physical speed of the cache memory (represented by vertical orange arrows). The right subfigure assumes that some new material/technology/effect decreases access time to one tenth of the time assumed on the left subfigure. In the figure, the technology (at considerable expenses) improved physical access time by a factor of ten, but the apparent access speed has improved only by a factor of less than two. *Even if the physical cache time could be reduced to zero, the apparent access time cannot be reduced below the time defined by the respective distances/interaction times. Mimicking the biology is useful also here: the time window, where the decision is made, is of the same size, independently*

of the path traversed by the signal (the axon length) and the speed of the signal (conduction velocity); and is in the order of the ‘processing time’ of the neurons.⁸

A recently proposed idea is to replace slow digital processing with quick analog processing [21, 22], and *may be proposed using any future new physical effect and/or material*, such as in [23]: they decrease T_p , but to make them useful for computing, their in-component transmission time T_t , and especially inter-component transmission time must be considerably decreased. *Neglecting their temporal behavior limits the utility of any new method, material or technology, if they are designed/developed/used in the spirit of the old (timeless) paradigm.*

3 Identifying Bottlenecks of Computing Due to Their Technical Implementation

3.1 Synchronous and Asynchronous Operation

The case depicted in Fig. 1a is an asynchronous operation: when the light cone arrives at the observer, the second processing can start. If we have additional observers, their T_t^A and T_t^B may be different, and we have no way to synchronize their operation. If we have another observer at the point mirrored to the origo, the light cone arrives at it at about the same T_t^A , but to synchronize the operation of the two observers, we would need $T_{synch} = T_t + T_t^A + T_t^B$. Instead, we issue another light cone (a central clock) at the origo (in the case of that light cone, the processing time is zero, just a rising edge) and observers are instructed to start their processing when this synchronizing light cone reaches their point of observation.

In the *time-space system*, not only observers on the surface of the cone, but also the ones inside the cone, can notice that the first light went on. If T_{synch} is large enough, all observers will notice the first light. After noticing the light, they all can start their processing at that time $t = 2 * T_p + T_{synch}$. Given that both $T_{p,i}$ and $T_{t,i}$ can be different, $T_{synch} \geq T_{p,i} + T_{t,i}$, for any observer i , must be fulfilled. This time is larger than any of the $T_{p,i} + T_{t,i}$ times: *for the rest of observers, the idle time increases*. Given that their internal wiring can be very different, we must choose the clock period according to the “worst-case”. *For the rest of observers, this constraint means a significant increase in their value $T_{t,i}$* . All observers must wait for the slowest one. The more observers (and the more steps!), the more waiting. This effect is considerable even inside the chip (at \leq cm distances); in the case of supercomputers, the distance is about 100 m.

A careful analysis [17] discovered that using synchronous computing (using clock signals) has a significant effect on performance of large-scale systems mimicking neuromorphic operation. The performance analysis [25] of large-scale brain

⁸The biology can change the conduction velocity, that needs energy, so finding an optimum is not as simple.

simulation facilities demonstrated an exciting parallel between modern science and large-scale computing. The commonly used 1 ms integration time, limited both the many-thread software (SW) simulator, running on general-purpose supercomputers, and the purpose-build HW brain simulator, to the *same value of payload performance*. Similar shall be the case very soon in connection with building the targeted large-scale neuromorphic systems, despite the initial success of specialized neuronal chips (such as [26, 27]). Although at a higher value (about two orders of magnitude higher than the one in [25]), systems built from such chips also shall stall because of the “*quantal nature of time*” [16], although using asynchronous operating mode can slighlyly rearrange the scene.

3.2 The High Speed Serial Bus

Components of technical computing systems (including biology-mimicking neuromorphic ones) are connected through a set of wires, called “bus”. The bus is essentially the physical appearance of the “technical implementation” of communication, stemming from the SPA, as illustrated in Fig. 4. The inset shows a simple neuromorphic use case: one input neuron and one output neuron communicating through a hidden layer, comprising only two neurons. Figure 4a mostly shows *the biological implementation: all neurons are directly wired to their partners*, i.e., a system of “parallel buses” (axons) exists. Notice that the operating time also comprises two non-payload times (T_i): data input and data output, which coincide with the non-payload time of the other communication party. The diagram displays logical and temporal dependencies of the neuronal functionality. The payload operation (“the computing”) can only start after data is delivered (by the, from this point of view, non-payload functionality: input-side communication), and output communication can only begin when the computing finished. Importantly, *communication and calculation mutually block each other*. Two important points that neuromorphic systems must mimic noticed immediately: i/ *the communication time is an integral part of the total execution time*, and ii/ *the ability to communicate is a native functionality of the system*. In such a parallel implementation, *performance of the system*, measured as the resulting total time (processing + transmitting), *scales linearly with increasing both non-payload communication speed and payload processing speed*.

Figure 4b shows a *technical implementation of a high-speed shared bus* for communication. To the right of the grid, the activity that loads the bus at the given time is shown. A double arrow illustrates communication bandwidth, the length of which is proportional to the number of packages the bus can deliver in a given time unit. We assume that the input neuron can send its information in a single message to the hidden layer, furthermore, that the processing by neurons in the hidden layer both starts and ends at the same time. However, the neurons must compete for accessing the bus, and only one of them can send its message immediately, the other(s) must wait until the bus gets released. The output neuron can only receive the

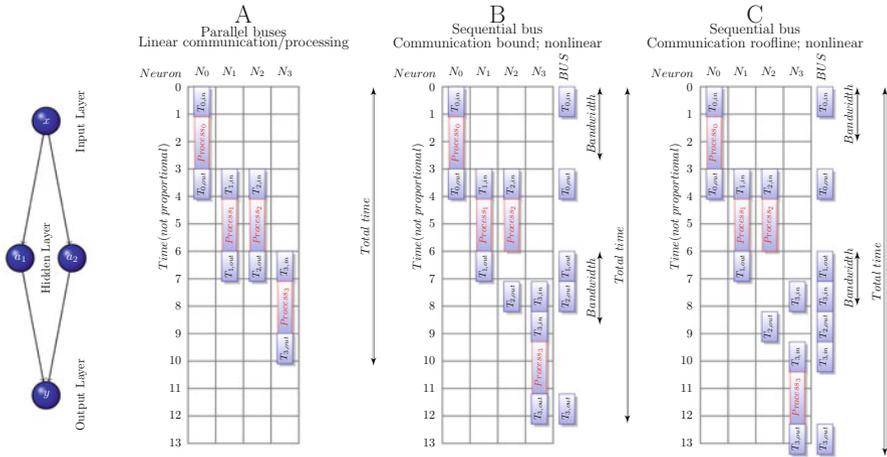


Fig. 4 Implementing neuronal communication in different technical approaches. (a): the parallel bus; (b) and (c): the shared serial bus, before and after reaching the communication “roofline” [24]

message when the first neuron completed it. Furthermore, the output neuron must first acquire the second message from the bus, and the processing can only begin after having both input arguments. *This constraint results in sequential bus delays both during non-payload processing in the hidden layer and payload processing in the output neuron.* Adding one more neuron to the layer, introduces one more delay.

Using the formalism introduced above, $T_i = 2 \cdot T_B + T_d + X$, i.e., the bus must be reached in time T_B (not only the operand delivered to the bus, but also waiting for arbitration: the right to use the bus), twice, plus the physical delivery through the bus. The X denotes “foreign contribution”: if the bus is not dedicated for “neurons in this layer only”, any other traffic also loads the bus: both messages from different layers and the general system messages may make processing slower.

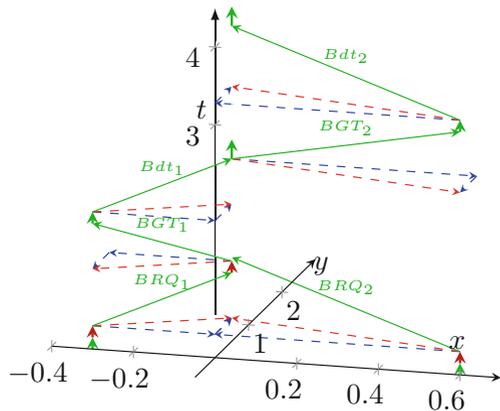
Even if only one single neuron exists in the hidden layer, it must use the mechanisms of sharing the bus, case by case. The physical delivery to the bus takes more time than a transfer to a neighboring neuron (both the arbiter and the bus are in cm distance range). If we have more neurons (such as a hidden layer) on the bus, and they work in parallel, they all must wait for the bus. The high-speed bus is very slightly loaded when only a couple of neurons are present, and its load increases linearly with the number of neurons in the hidden layer (or, maybe, all neurons in the system). The temporal behavior of the bus, however, is different. Under the biology-mimicking workload, the second neuron must wait for all its inputs originating in the hidden layer. If we have L neurons in the hidden layer, the transmission time of the neuron behind the hidden layer is $T_i = L \cdot 2 \cdot T_B + T_d + X$. This temporal behavior explains why “*shallow networks with many neurons per layer ... scale worse than deep networks with less neurons*” [10]: the physical bus delivery time T_d , as well as the processing time T_p , become marginal if the layer

forces to make many arbitrations to reach the bus: the number of the neurons in the hidden layer defines the transfer time (Recall Fig. 1a for the consequences of increasing the transfer time). In deeper networks, the system sends its messages at different times in different layers (and, even they may have independent buses between the layers), although *the shared bus persists in limiting the communication*. Notice that there is no way to organize the message traffic: only one bus exists.

Figure 5 discusses, in terms of “temporal logic”, the case depicted in the inset in Fig. 4 (where the same operation is discussed in conventional terms): why using high-speed buses for connecting modern computer components leads to very severe performance loss, especially when one attempts to imitate neuromorphic operation. The two neurons of the hidden layer are positioned at $(-0.3,0)$ and $(0.6,0)$. The bus is at position $(0,0.5)$. The two neurons make their computation (green arrows at the position of neurons), then they want to tell their result to fellow neurons. Unlike in biology, first they must have access to the shared bus (red arrows). Core at $(-0.3,0)$ is closer to the bus, so its request is granted. As soon as the grant signal reaches requesting core, the bus operation is initiated, and the data starts to travel to the bus. As soon as it reaches the bus, it is forwarded by the high speed of the bus, and at that point bus request of the other core is granted, and finally, also calculated result of the second neuron is bused.

At this point comes into picture the role of the workload on the system: the two neurons in the hidden layer want to use the single shared bus, at the same time, for communication. As a consequence, *the apparent processing time is several times higher, than the physical processing time, and it increases linearly with the number of neurons in the hidden layer* (and maybe with also the total number of neurons in the system, if a single high-speed bus is used). *In vast systems, especially when attempting to mimic neuromorphic workload, the speed of the bus is getting marginal*. Notice that times shown in the figure are not proportional: the (temporal) distances between cores are in the several picoseconds range, while the bus (and the arbiter) are at a distance well above nanoseconds, so *the actual temporal behavior (and the idle time stemming from it) is much worse than the figure suggests*. This

Fig. 5 The operation of the sequential bus, in time-space coordinate system system. Near to axis t , the lack of vertical arrows signals “idle waiting” time



is why “The idea of using the popular shared bus to implement the communication medium is no longer acceptable, mainly due to its high contention” [28]. The figure suggests to use another design principle instead of using the bus exclusively, directly from the position of the computing component (Fig. 5).

Given that the bus bandwidth is finite, there comes the point when the amount of messages exceeds the available bus bandwidth. Figure 4c demonstrates the case, where for better visibility, the bus bandwidth is lower, but the required packet bandwidth slice is the same. In this case, the second neuron in the hidden layer cannot send its message when the first one finishes its transmission: the bus transmission roofline [24] is reached. In that case the transmission time shall be extended with a new term $T_t = (B + L) \cdot 2 \cdot T_B + T_d + X$, where B is the number of messages above the number of messages that the bus can deliver in a unit time. Reaching the roofline causes further extra delay in both non-payload and payload processing times, extending the total execution time. A single sequential bus can deliver messages only one after the other, i.e., *increasing number of neurons increases utilization of the bus and prolongs total execution time as well as apparent processing time of the individual neurons*. This effect can be so strong in large systems, that emergency measures must have been introduced: the events “are processed as they come in and are dropped if the receiving process is busy over several delivery cycles” [25].

When using a shared bus, increasing either processing speed or communication speed does not affect linearly the total execution time any more. Furthermore, it is not the bus speed that limits performance. Recall Fig. 1a again, to see, how the time projection of a relatively small increase in the transfer time T_t can lead to a relatively large change in the value of apparent processing time T_A ; and so

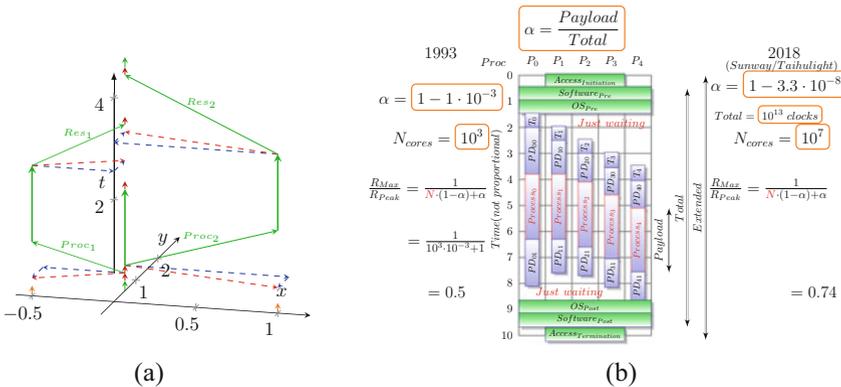


Fig. 6 The parallelized sequential operation as described in the proposed time-space system, with its simplified, non-technical model [6]. (a) Time diagram of parallelized sequential operation in time-space. (a) A non-technical, simplified model of parallelized sequential computing operations. Notice the different nature of those contributions and that they have only one common feature: *they all consume time*. The vertical scale displays the actual activity for processing units shown on the horizontal scale

leads to incomprehensible slowdown of the system: the slowest component defines efficiency. Conventional way of communication may work fine as long as there is no competition for the bus, but leads to queuing of messages in the case of (more than one!) independent sources of communication. The bursty nature, caused by the need of central synchronization, tops the effect, and leads to a “communicational collapse” [29], that denies huge many-processor systems, especially neuromorphic ones [30].

To have a chance to connect a large number of computing units in biology-mimicking systems, drastically new bus system and drastically new traffic organization is required [31]. Using a single high-speed bus greatly contributes to the experienced very low efficiency of Artificial Neural Network (ANN)s [5], and finally that “Core progress in AI has stalled in some fields” [32].

3.3 Parallelized Sequential Processing

Present technical approaches assume a linear dependence between payload and nominal performances of computing systems as “*Gustafson’s formulation [7] gives an illusion that as if N [the number of the processors] can increase indefinitely*” [33]. The fact that “*in practice, for several applications, the fraction of the serial part happens to be very, very small thus leading to near-linear speedups*” [33] (see also value of α in Fig. 1b), however, misled the researchers. *Gustafson’s “linear scaling” neglects all non-payload contributions entirely, including the temporal behavior of the components.* He established his conclusions on only several hundred processors. The interplay of improving parallelization and general HW development (including non-determinism of modern HW [34]) covered for decades that *weak scaling was used far outside of its range of validity [5]*. In our terminology, Gustafson’s assumption means that $T_t = 0$, which is not the case, in any computing system, and especially not in the case of neuromorphic computing systems. As pointed out above, having idle time in computing systems is inevitable; the vastly increased number of idle cycles due to physical size and operating mode of computing systems led to the effects detailed above.

Figure 6a depicts temporal diagram of distributed parallel processing in the introduced time-space system. One of the PUs (in our case the one at (0,0.5)) orchestrates the operation, including receiving the start command and returning the result. This core makes some sequential operations (such as initializing data structures, short green arrow), then it sends the start command and operands to fellow cores at (−0.5,0) and (1,0), one after the other. Signal propagation takes time (depending on distance from the coordinator), and after that time, fellow cores can start their calculation (their part of the parallelized portion). Of course, orchestrator PU must start all fellow PUs (red arrows), then it can start its portion of distributed processing. In the case of large number of fellow processors, it may be advantageous

if the coordinator does not have its own portion of parallelizable code:⁹ executing that code may delay receiving results from the fellow processors.

As fellow PUs finish their portion, they must transmit their data Res_i to the orchestrator, that receives those data in *sequential* mode, and finally makes some closing *sequential* processing. Again, the *inherently sequential-only portion* [35] of the task increases with number of cores and its *idle waiting time* (time delay of signals) increases with physical size (cable length). The times shown in the figure are not proportional, and largely depend on type of the system. For example, in supercomputers, total calculation time is in the hours range, number of red arrows (without clustering) can be up to several millions, and the delay, due to the finite speed of signal propagation, in several thousand clock cycles (in the case of using Ethernet networks, several millions).

Notice, that the figures assume no dependence (such as logical dependence on sharing physical resources) between the computing objects (threads), and especially not the case when several SW threads share the same PU. Notice also, that the orchestrating PU must wait results from all fellow PUs, i.e. *the slowest branch defines performance*.

Amdahl listed [36] different reasons why losses in “computational load” can occur. Fortunately, Amdahl’s idea enables us to *put everything that cannot be parallelized, i.e., distributed between the fellow processing units, into the sequential-only fraction*. For describing the parallel operation of sequentially working units, the model depicted in Fig. 6b was prepared. The technical implementations of the different parallelization methods show up virtually infinite variety [37], so we present here a (by intention) strongly simplified, non-technical, model. The model has some obvious limitations, among others, because of the non-determinism of modern HW systems [34, 38].

In addition to “idle time”’s discussed above, the serialized parallel processing adds one more contribution. Even the simplest (parallelized sequential) task has a non-parallelizable *portion of time*, that—according to Amdahl’s Law—limits the achievable payload computing performance. Here the *sequential bus* and the *transmission delay* play a role, again. Because, in the SPA, the initiating processor can address only one processor (or through clustering: only a few of them), the other processors must make additional *idle waiting*: the loop to address them takes time, and the cable length significantly increases their T_i . This effect, however, comes to light only at a relatively high number of cores *and* real-life workloads. At a lower number of cores *and* HPL-class benchmarks, only a slight deviation from the linearity, predicted by the “weak scaling”, can be noticed.

The right subfigure in Fig. 7 displays the payload performance of a many-processor SPA system when executing different workloads (that define the non-payload to payload ratio); for the math details see [6, 16]. The top diagram lines represent the best payload performance that the supercomputers can achieve when running the benchmark HPL, which represents the minimum communication

⁹For examples see the architectures of supercomputers *Taihulight* and *Summit*.

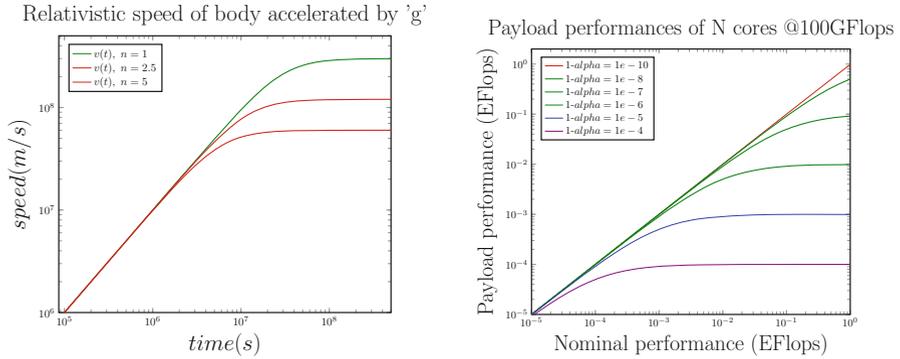


Fig. 7 The limiting effect considered in the “modern” theories. One left side, the speed limit, as explained by the theory of relativity, is illustrated. The refractory index of the medium defines the value of the speed limit. On the right side, the payload performance limit of the parallelized sequential computing systems, as explained by the “modern paradigm”, is illustrated. The ratio of the non-payload to payload processing defines the value of the payload performance

a parallelized sequential system needs. *The bottom diagram line represents the estimation of the payload performance that neuromorphic-type processing can achieve in SPA systems (See also Fig. 1b).* Notice the similarity with the left subfigure: *under extreme conditions, in the science, an environment-dependent speed limit exist, and in computing, a workload-dependent payload performance limit exists [16].*

To have hopes to significantly increase computing performance of our present cutting-edge conventional and future neuromorphic computing systems, principle other than parallelizing otherwise sequentially processing systems, must be discovered. The recent paradigm leads to not only inherent performance limits, but also to irrationally high power consumption.

3.4 Communication

One of the worst computing performance limiting factors is the method of communication between processors, which increases exponentially with increasing complexity/number. Historically, in the model of computing proposed by von Neumann, there was one single entity, an isolated (non-communicating) processor, whereas in bio-inspired models, billions of entities, organized into specific assemblies, cooperate via communication. (Communication here means not only sending data, but also sending/receiving signals, including synchronization of the operation of entities.) Neuromorphic systems, expected to perform tasks in one paradigm, but assembled from components manufactured using principles of (and implemented by experts trained in) the other paradigm, are unable to perform at the required speed and efficacy for real-world solutions. The larger the system, the higher the communication load and the performance debt. With reference to Fig. 1a,

time contribution of the communication is part of the processing time T_p , although the overwhelming part of it could be done in parallel with the computing activity. This feature both decreases available processing capacity of a neuron, and strongly changes value of R . More importantly, it must use communication facilities through Input/Output (I/O) instructions, wasting a massive amount of time for that.

4 The Effect of Temporal Behavior on Scaling

Dependence of *payload* performance on *nominal* performance in many-many processor systems is strongly nonlinear at higher performance values (implemented using a large number of processors). This effect is especially disadvantageous for networks, such as neuromorphic ones, that show up non-proportionally much idle wait time, mainly because of the reasons presented above. The linear dependence at low nominal performance values explains why initial successes of *any new technology, material or method* in the field, using the classic computing model, can be misleading: in simple cases classic paradigm performs tolerably well thanks to that *compared to biological neural networks, current neuron/dendrite models are simple, the networks small and learning models appear to be rather basic.*

The biology is aware of that the transmission time is a crucial part of the processing. *“Importantly, distally projecting axons of long-range interneurons have several-fold thicker axons and larger diameter myelin sheaths than do pyramidal cells, allowing for considerably faster axon conduction velocity”* [39]. Faster conduction increases the energy consumption of a cell (needing more myelin), but it prevents a race condition between the signals. The biology “wastes” extra energy only when required, and here there appears the need to refine the “fire and wire together” operating principle with modulating the conduction velocity. The surprising resemblance between Fig. 8a and Fig. 7 in [39] also underlines the importance of making a clear distinction between handling ‘near’ and ‘far’ signals. Although the Inter-Core Communication Block (ICCB) blocks in the biology-mimicking architecture [31] can adequately represent ‘locally connected’ interneurons and the ‘G’ gateway the ‘long-range interneurons’, the biological conduction time must be separately maintained. Computer technology cannot speed up communication selectively, as biology does, and it is not worth to slow it down selectively. Making time-stamps and relying on computer network delivery principles is not sufficient: *temporal behavior is a vital feature of biology-mimicking systems and we must not replace them with synchronization principles of computing.*

5 Summary

Statements such as *“The von Neumann architecture is fundamentally inefficient and non-scalable for representing massively interconnected neural networks”* [40]

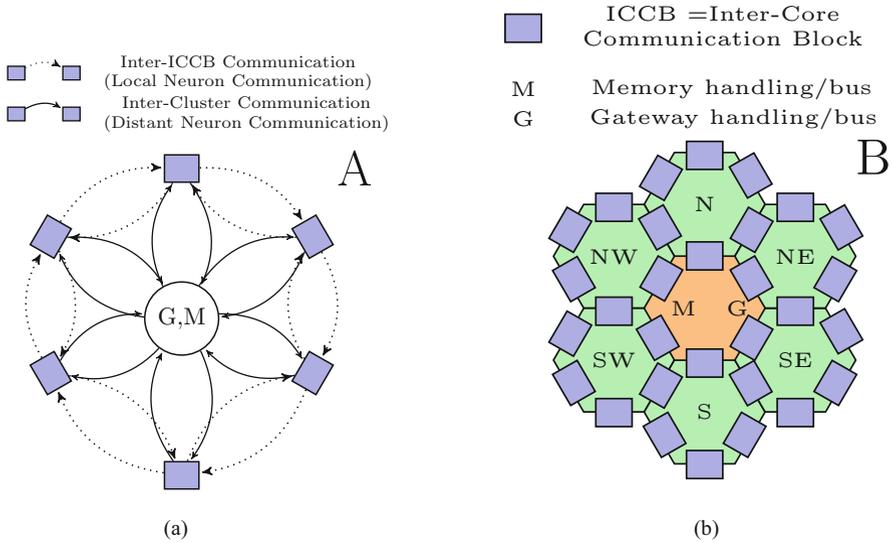


Fig. 8 The communication scheme between local and farther neurons, as can be implemented in technically [31]. (a) The conceptual communication diagram (compare to Fig. 7 in [39]), mimicking the communication between local neurons the farther neurons. (b) The proposed implementation: the Inter-Core Communication Blocks represent a “local bus” (directly wired, with no contention), while the cores can communicate with the cores in other clusters through the ‘G’ gateway as well as the ‘M’ (local and global) memory

should be modified like this “*the architectures based on the non-temporal abstraction proposed by von Neumann*”. Especially the figures above, provide a very clear pointer: *to make efficient and large systems (including neuromorphic ones), the fundamental principles of operation of computing, communication, including the bus system and principle of handling messages, as well as the cooperation between processors, must be scrutinized and drastically changed.* Comprehending the timely behavior of the components can serve as a good starting point to do so.

References

1. K. Asanovic, R. Bodik, J. Demmel, T. Keaveny, K. Keutzer, J. Kubiatowicz, N. Morgan, D. Patterson, K. Sen, J. Wawrzynek, D. Wessel, K. Yelick, A view of the parallel computing landscape. *Commun. ACM* **52**(10), 56–67 (2009)
2. US National Research Council, The Future of Computing Performance: Game Over or Next Level? (2011). [Online]. Available: <http://science.energy.gov/~media/ascr/ascac/pdf/meetings/mar11/Yelick.pdf>
3. I. Markov, Limits on fundamental limits to computation. *Nature* **512**(7513), 147–154 (2014)
4. J.P. Singh, J.L. Hennessy, A. Gupta, Scaling parallel programs for multiprocessors: Methodology and examples. *Computer* **26**(7), 42–50 (1993)

5. J. Végh, Which scaling rule applies to Artificial Neural Networks, in *Computational Intelligence (CSCI) The 22nd Int'l Conf on Artificial Intelligence (ICAI'20)* (IEEE, 2020). Accepted ICA2246, in print. [Online]. Available: <http://arxiv.org/abs/2005.08942>
6. J. Végh, Finally, how many efficiencies the supercomputers have? J. Supercomput. (2020). [Online]. Available: <https://doi.org/10.1007%2Fs11227-020-03210-4>
7. J.L. Gustafson, Reevaluating Amdahl's law. *Commun. ACM* **31**(5), 532–533 (1988)
8. C. Liu, G. Bellec, B. Vogginger, D. Kappel, J. Partzsch, F. Neumäker, S. Höppner, W. Maass, S.B. Furber, R. Legenstein, C.G. Mayr, Memory-efficient deep learning on a SpiNNaker 2 prototype. *Frontiers Neurosci.* **12**, 840 (2018). [Online]. Available: <https://www.frontiersin.org/article/10.3389/fnins.2018.00840>
9. Top500.org, Retooled Aurora Supercomputer Will Be America's First Exascale System (2017). <https://www.top500.org/news/retooled-aurora-supercomputer-will-be-americas-first-exascale-system/>
10. J. Keuper, F.-J. Preundt, Distributed training of deep neural networks: Theoretical and practical limits of parallel scalability, in *2nd Workshop on Machine Learning in HPC Environments (MLHPC)* (IEEE, 2016), pp. 1469–1476. [Online]. Available: <https://www.researchgate.net/publication/308457837>
11. J. Végh, How deep the machine learning can be, ser. *A Closer Look at Convolutional Neural Networks* (Nova, In press, 2020), pp. 141–169. [Online]. Available: <https://arxiv.org/abs/2005.00872>
12. US DOE Office of Science, Report of a Roundtable Convened to Consider Neuromorphic Computing Basic Research Needs (2015). https://science.osti.gov/-/media/ascr/pdf/programdocuments/docs/Neuromorphic-Computing-Report_FNLBLP.pdf
13. G. Bell, D.H. Bailey, J. Dongarra, A.H. Karp, K. Walsh, A look back on 30 years of the Gordon Bell Prize. *Int. J. High Performance Comput. Appl.* **31**(6), 469–484 (2017). [Online]. Available: <https://doi.org/10.1177/1094342017738610>
14. S(o)OS project, Resource-independent execution support on exa-scale systems (2010). <http://www.soos-project.eu/index.php/related-initiatives>
15. Machine Intelligence Research Institute, Erik DeBenedictis on supercomputing (2014). [Online]. Available: <https://intelligence.org/2014/04/03/erik-debenedictis/>
16. J. Végh, A. Tisan, The need for modern computing paradigm: Science applied to computing, in *Computational Science and Computational Intelligence CSCI The 25th Int'l Conf on Parallel and Distributed Processing Techniques and Applications* (IEEE, 2019), pp. 1523–1532. [Online]. Available: <http://arxiv.org/abs/1908.02651>
17. J. Végh, How Amdahl's Law limits the performance of large artificial neural networks. *Brain Informatics* **6**, 1–11 (2019). [Online]. Available: <https://braininformatics.springeropen.com/articles/10.1186/s40708-019-0097-2/metrics>
18. R. Hameed, W. Qadeer, M. Wachs, O. Azizi, A. Solomatnikov, B.C. Lee, S. Richardson, C. Kozyrakis, M. Horowitz, Understanding sources of inefficiency in general-purpose chips, in *Proceedings of the 37th Annual International Symposium on Computer Architecture*, ser. ISCA '10 (ACM, New York, NY, USA, 2010), pp. 37–47. [Online]. Available: <http://doi.acm.org/10.1145/1815961.1815968>
19. A. Haidar, P. Wu, S. Tomov, J. Dongarra, Investigating half precision arithmetic to accelerate dense linear system solvers, in *Proceedings of the 8th Workshop on Latest Advances in Scalable Algorithms for Large-Scale Systems*, ser. ScalA '17 (ACM, New York, NY, USA, 2017), pp. 10:1–10:8
20. J. Backus, Can programming languages be liberated from the von Neumann Style? A functional style and its algebra of programs. *Commun. ACM* **21**, 613–641 (1978)
21. E. Chicca, G. Indiveri, A recipe for creating ideal hybrid memristive-CMOS neuromorphic processing systems. *Appl. Phys. Lett.* **116**(12), 120501 (2020). [Online]. Available: <https://doi.org/10.1063/1.5142089>
22. Building brain-inspired computing. *Nature Communications* **10**(12), 4838 (2019). [Online]. Available: <https://doi.org/10.1038/s41467-019-12521-x>

23. P. Cadareanu, et al., Rebooting our computing models, in *Proceedings of the 2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)* (IEEE Press, 2019), pp. 1469–1476
24. S. Williams, A. Waterman, D. Patterson, Roofline: An insightful visual performance model for multicore architectures. *Commun. ACM* **52**(4), 65–76 (2009)
25. S.J. van Albada, A.G. Rowley, J. Senk, M. Hopkins, M. Schmidt, A.B. Stokes, D.R. Lester, M. Diesmann, S.B. Furber, Performance comparison of the digital neuromorphic hardware SpiNNaker and the neural network simulation software NEST for a full-scale cortical microcircuit model. *Frontiers Neurosci.* **12**, 291 (2018)
26. F. Akopyan, Design and tool flow of IBM’s TrueNorth: An ultra-low power programmable neurosynaptic chip with 1 million neurons, in *Proceedings of the 2016 on International Symposium on Physical Design*, ser. ISPD ’16 (ACM, New York, NY, USA, 2016), pp. 59–60. [Online]. Available: <http://doi.acm.org/10.1145/2872334.2878629>
27. M. Davies, et al, Loihi: A neuromorphic manycore processor with on-chip learning. *IEEE Micro* **38**, 82–99 (2018)
28. L. de Macedo Mourelle, N. Nedjah, F.G. Pessanha, *Reconfigurable and Adaptive Computing: Theory and Applications*, ch. 5: Interprocess Communication via Crossbar for Shared Memory Systems-on-chip (CRC press, 2016)
29. S. Moradi, R. Manohar, The impact of on-chip communication on memory technologies for neuromorphic systems. *J. Phys. D Appl. Phys.* **52**(1), 014003 (2018)
30. S.B. Furber, D.R. Lester, L.A. Plana, J.D. Garside, E. Painkras, S. Temple, A.D. Brown, Overview of the SpiNNaker system architecture. *IEEE Trans. Comput.* **62**(12), 2454–2467 (2013)
31. J. Végh, How to extend the Single-Processor Paradigm to the Explicitly Many-Processor Approach, in *2020 CSCE, Fundamentals of Computing Science* (IEEE, 2020). Accepted FCS2243, in print. [Online]. Available: <https://arxiv.org/abs/2006.00532>
32. M. Hutson, Core progress in AI has stalled in some fields. *Science* **368**, 6494/927 (2020)
33. Y. Shi, Reevaluating Amdahl’s Law and Gustafson’s Law (1996). https://www.researchgate.net/publication/228367369_Reevaluating_Amdahl's_Law_and_Gustafson's_Law
34. V. Weaver, D. Terpstra, S. Moore, Non-determinism and overcount on modern hardware performance counter implementations, in *2013 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS)*, pp. 215–224 (April 2013)
35. F. Ellen, D. Hendler, N. Shavit, On the inherent sequentiality of concurrent objects. *SIAM J. Comput.* **43**(3), 519–536 (2012)
36. G.M. Amdahl, Validity of the single processor approach to achieving large-scale computing capabilities,” in *AFIPS Conference Proceedings*, vol. 30, pp. 483–485 (1967)
37. K. Hwang, N. Jotwani, *Advanced Computer Architecture: Parallelism, Scalability, Programmability*, 3rd edn. (McGraw Hill, 2016)
38. P. Molnár, J. Végh, Measuring performance of processor instructions and operating system services in soft processor based systems, in *18th Internat. Carpathian Control Conf. ICC*, pp. 381–387 (2017)
39. G. Buzsáki, X.-J. Wang, Mechanisms of gamma oscillations. *Ann. Rev. Neurosci.* **3**(4), 19:1–19:29 (2012)
40. J. Sawada et al., TrueNorth ecosystem for brain-inspired computing: Scalable systems, software, and applications, in *SC ’16: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 130–141 (2016)

Evaluation of Classical Data Structures in the Java Collections Framework



Anil L. Pereira

1 Introduction

The Java Collections framework [1] is a unified architecture for representing and manipulating data collections. The classical data structures [2] implemented in the Java Collections framework and considered in this paper are array, array list, linked list, doubly linked list, stack, and queue [3]. This paper asks an important question and attempts to answer it. The question is, what are the important performance considerations of the classical data structures as implemented in the Java Collections framework when using asymptotic analysis [4] for software design? For example, inserting or removing an element at the end of an array list or linked list data structure takes constant time irrespective of the number of elements in the data structure. However, the software execution time relative to the array list is much faster due to its memory being allocated contiguously and with less overhead. Even though the time complexity in this case is the same for both data structures, clearly the array list would be a better choice among the two in order to buffer small amounts of data while transmitting or receiving data at high speeds through a network. The paper seeks to answer the above question by analyzing the performance gap between the data structures when similar operations have equal time and equal space complexity. To the best of the author's knowledge, there is no work reported in the available technical literature that poses the above question and attempts to answer it. Why is this question important? It is important because the performance of software applications for computer networking, Web services, and cloud computing, with respect to speed, scalability, fault tolerance, and quality of service, is critical. Designing software for these applications involves choosing

A. L. Pereira (✉)

School of Science and Technology, Georgia Gwinnett College, Lawrenceville, GA, USA
e-mail: apereira@ggc.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_34

493

the right kind of data structure. Choosing the right kind of data structure is crucial because its performance with respect to space (memory utilization, i.e., how much memory is used to store data and what is the overhead) and performance of its operations with respect to time (execution speed, i.e., how fast does the software implementation run) play a significant role in determining the overall performance of the application.

Software developers should know how to compare various data structures based on their memory utilization and performance of their operations. As with most choices in computer programming and design, no method is well suited to all circumstances. A linked list data structure might work well in one case, but cause problems in another case. Also, how does the performance of one data structure scale with data size compared to another data structure? To answer this question, software developers can use asymptotic analysis for a theoretical comparison between the data structures regarding the scalability of their performance. However, software developers should be aware of any practical considerations affecting scalability of performance that may arise from the implementation of the data structures and their operations. They should be able to identify any implementation overhead that may adversely affect practical performance, for example, using data types incorrectly (using double where byte would suffice) or excessive recursion where iteration might be used. In this paper, improvements are proposed to obtain better performance than currently available. The required data for performance evaluation was obtained through software implementation conducted in Java. For the software implementation, Java methods to profile available program memory and execution time of operations were used.

The broader impacts of this work can be in academia and research. The Java Collections framework is increasingly used in undergraduate computer science and information technology courses covering data structures. Students can experimentally verify the practical performance of the data structures using the performance evaluation method described in this paper. Researchers can explore and possibly improve the implementation of data structures in similar frameworks of other programming languages.

The paper is organized as follows. Section 2 contains an explanation of data structures. Section 3 discusses array lists and the asymptotic analysis of their operations. Section 4 discusses linked lists and the asymptotic analysis of their operations and compares them to array lists. Section 5 discusses doubly linked lists and the asymptotic analysis of their operations. Section 6 contains performance evaluation. Section 7 explains stacks and discusses how best to implement them. Section 8 explains queues and discusses how best to implement them. Section 9 contains conclusions and future work.

2 Data Structure

A data structure is an organized collection of data. A data structure not only stores data but also supports the operations for manipulating data in the structure. For example, a classical data structure, array list, holds a collection of data in sequential order and is dynamically resizable (its capacity can increase or decrease to accommodate the amount of data). You can find the size of the array list and store, retrieve, delete, and modify data in the array list. Other examples of classical data structures are arrays, lists, stacks, and queues. A list is a collection of data stored sequentially. Insertion and deletion operations are supported anywhere in the list. A stack can be perceived as a special type of list where insertions and deletions take place only at one end, referred to as the top of the stack. A queue represents a waiting list, where insertions take place at the back (also referred to as the tail) of the queue and deletions take place from the front (also referred to as the head) of a queue.

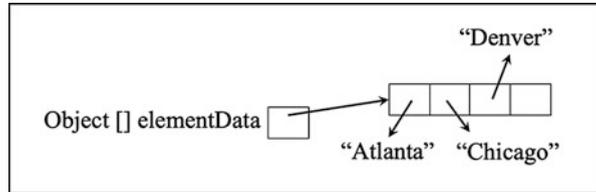
3 Array List

To explain an array list and its limitations, we will first see how the Java program code implementing it compiles and executes. Instantiating a generic class by passing a specific type (e.g., String) to the parameterized type is called generic type invocation, as shown in the first line of the Java program code below.

```
ArrayList <String> list = new ArrayList<>(4);
list.add("Atlanta");
list.add("Chicago");
list.add("Denver");
ArrayList list = new ArrayList();
list.add("Atlanta");
list.add("Chicago");
list.add("Denver");
```

Because, the class `ArrayList` is instantiated with the generic type invocation of `String`, the compiler first checks if only `String` objects are added to the array list. If any other object, for example, `Integer`, is added, then there will be a compile time error. Also, `String` is declared final, meaning it cannot be extended, and thus cannot have subclasses. A class would be made final if the programmer desires that none of its methods be overridden, so that only those specific behaviors that are desired by the programmer are maintained. But, an array list of a class that is not declared final, can contain objects of the subclasses, because of the compiler support for Polymorphism [5]. In the next step, as shown in the fifth line of code above, the compiler performs type erasure. Meaning, the parameterized type is removed. This can be done because at this point only objects of the class or subclass of the generic type invocation would be stored in the array list. The array list has an instance variable named `elementData` that can reference an array of instances of the class `Object`, as shown in Fig. 1. The class `Object` is the root of the hierarchical class tree

Fig. 1 Array list implemented using an array of the class Object



in Java. In other words, it is the superclass of all the classes. When class `ArrayList` is instantiated with generic type invocation, an array of `Object` is also instantiated. Objects of the generic type invocation that are added to the array list are stored in the array.

The performance of the `add` (or `insert`) operation, that is, adding (or inserting) an object to the array list, depends upon where in the underlying array the object is being added. If adding objects to the end of an existing sequence of objects (best case), one after another, then it takes constant time to add an object no matter how many objects there are in the array and how many objects are added because the array is indexed. Indexes allow for direct access (also called random access) to memory. This means that it does not matter where in memory you are accessing or storing an object or how many objects are accessed or stored, it takes the same time to access or store a single object, i.e., constant time.

The constant time to add is represented by a time complexity of Big-Theta (1), symbolically noted as $\Theta(1)$. Big-Theta notation is used in asymptotic analysis. Asymptotic analysis refers to the study of an algorithm's or operation's performance with respect to resource usage (time complexity, i.e., execution time, and space complexity, i.e., memory size) as the data size grows larger. Asymptotic analysis provides a simplified model of resource usage of an algorithm or operation. Asymptotic notation shows how an algorithm or operation scales when compared to another algorithm or operation. In other words, it shows the rate of growth of cost of an algorithm or operation with respect to time or space as n (the data size grows). Θ is used to indicate that the upper and lower bounds for the cost of an operation are the same within a constant factor.

3.1 Limitations of Array List

If the array list is full, then the elements have to be copied to a new bigger array, and the next element can then be added to the new array. This reallocation is done automatically in an array list. It may not be possible to reallocate if memory is fragmented. The cost of reallocation can be averaged out over many insertions, and the time complexity of an insertion due to reallocation would still be $\Theta(1)$.

Insertion at the Beginning of an Array List

For adding to the beginning of the array list (worst case), all elements must be shifted one place to the right before adding the element to the beginning of the array list. Reallocation may be required. The execution time increases linearly with the size of the array list. Asymptotically the time complexity is $\Theta(n)$.

Insertion at a Specified Index in an Array List

Before inserting a new element at a specified index, all the elements at and after the index must be shifted to the right one place and the list size must be increased by 1. On average, half the elements in the array list must be shifted to the right one place when adding (inserting) an element at a particular index. The execution time increases linearly with the size of the array list. Reallocation may be required. Asymptotically, the time complexity is also $\Theta(n)$. This is because, asymptotically $n/2$, the data size to be right shifted if adding in the middle (average case), or, n , the data size to be right shifted if adding at the beginning, does not matter. Linear increase or decrease of the data size (i.e., increase or decrease by a constant factor) does not affect the growth rate of the cost of an operation with respect to its execution time.

Deletion at a Specified Index

To remove an element at a specified index (average case for remove), all the elements after the index must be shifted to the left by one position, and the list size must be decreased by 1. On average, half the elements in the array list must be shifted to the left one place when removing (deleting) an element at a particular index. The execution time increases linearly with the size of the array list. The left shifts are necessary to avoid fragmentation in the array list. Fragmentation adversely affects iteration because the elements are no longer stored contiguously. An array from (which many elements are removed) may also have to be resized in order to avoid wasting too much space, though the cost of resizing can be averaged out over many deletions. Asymptotically, the time complexity is $\Theta(n)$.

Deletion at the Beginning

To remove from the beginning (worst case), all following elements must be shifted to the left one place. The execution time increases linearly with the size of the array list. An array from which many elements are removed may also have to be resized in order to avoid wasting too much space. Asymptotically, the time complexity is also $\Theta(n)$.

Deletion at the End

To remove from the end (best case), the reference to the last element can be replaced by a null pointer. This operation is done in constant time. An array from which many elements are removed may also have to be resized in order to avoid wasting too much space. Asymptotically, the time complexity is $\Theta(1)$.

Asymptotically $n/2$ (data size shifted if adding or removing from middle), n (data size shifted if adding or removing from beginning), or any other linear decrease (e.g., $n/3$, $n/4$, $n/5$, and so on) or increase (e.g., $2n$, $3n$, $4n$, and so on) does not matter. Linear increase or decrease of the data size (i.e., increase or decrease by a constant factor) does not affect the growth rate of the cost of an operation with respect to its execution time. Asymptotic notation of Big-Oh (O), Big-Omega (Ω), Big-Theta (Θ), small-Oh (o), and small-omega (ω) are not the same as the best, worst, or average case. For example, there is a difference between the best, worst, or average case and asymptotic notation like Big-Theta (Θ). For an array list, the best/worst/average case for the add (or insert) operation is addLast/addFirst/addMiddle. They are each $\Theta(1)/\Theta(n)/\Theta(n)$.

4 Linked List

A linked list, as shown in Fig. 2, consists of a chain of objects called nodes, each containing a data element and linked to its next neighbor via a pointer. A node can be defined as a class in Java. The Java class for a node consists of two variables, an element (generic) object reference to data and an object reference to the next neighboring node. Nodes can be non-contiguously stored in memory. The memory size due to storing the references to data and the next node can be considered overhead. Asymptotically, the space complexity of memory overhead in a linked list is $\Theta(n)$. The maximum size, i.e., the maximum number of nodes of a linked list, is constrained by the amount of available heap memory allocated to the program. A linked list is less susceptible to memory fragmentation than array list because nodes in a linked list do not have to be contiguously stored in memory, unlike the object references in an array list.

Data access and reading take longer in a linked list as the number of nodes increases, because on average, in order to access and read an element, all preceding

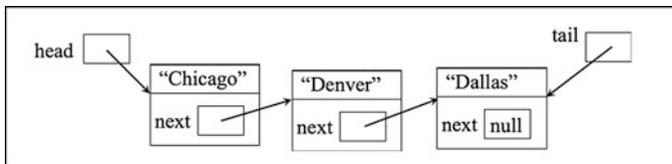


Fig. 2 Linked list of class String containing three nodes

nodes must be traversed to get to the particular node. Asymptotically, the time complexity to access and read a data element in a linked list is $\Theta(n)$.

An array list consists of object references pointing to the data. Object references in an array list are stored contiguously. Array lists require less memory than linked lists for the same number of data elements. Asymptotically, the space complexity of the memory overhead is also $\Theta(n)$. The maximum size, i.e., the maximum number of object references of an array list, is equal to the maximum positive value of the *int* data type, which is $(2^{31}-1)$. However, the maximum size is restricted practically by heap memory allocated to the program. An array list is more susceptible to memory fragmentation than a linked list due to the need for contiguous memory allocation, and in extreme cases reallocation to resize the array list may not be possible due to a memory block of sufficient size being unavailable.

Data access and reading in an array list is done in constant time, because an array list supports random access (direct access). An index (as shown in the Java program code below) is translated to a memory address which allows direct retrieval via the operating system and hardware. Asymptotically, the time complexity to access and read a data element in an array list is $\Theta(1)$.

```
list.get(0) => array[0] => "Atlanta"  
list.get(1) => array[1] => "Chicago"  
list.get(2) => array[2] => "Denver"
```

4.1 Insertion Operations for a Linked List

There are three implementations of the add operation: `addLast(E o)`, `addFirst(E o)`, and `add(int index, E o)`.

`addLast(E o)`: Creates a new node for the given element and adds the node to the end of the linked list. Takes constant time no matter how big the linked list, because a node needs to be added only at the end without displacing any other nodes before it. Asymptotically, the time complexity is $\Theta(1)$.

`addFirst(E o)`: Creates a new node for the given element and adds the node to the beginning of the linked list. Takes constant time no matter how big the linked list, because a node needs to be added only at the beginning without displacing other nodes after it. Asymptotically, the time complexity is $\Theta(1)$.

`add(int index, E o)`: Creates a new node for the given element and adds the node at a particular position (given by the index) in the linked list. On average, a linked list must be traversed along its nodes (beginning from the first node) in order to reach the point of insertion. Asymptotically, the time complexity is $\Theta(n)$. It takes constant time if a pointer to the last node inserted is maintained. This could be done if a sequence of nodes were inserted one after another. Asymptotically, the time complexity is $\Theta(1)$.

4.2 Deletion Operations for a Linked List

There are three implementations of the remove operation: `removeFirst()`, `removeLast()`, and `remove(int index)`.

`removeFirst()`: Removes the first node of the linked list and returns its reference to the calling program. Takes constant time no matter how big the linked list, because a node needs to be removed only at the beginning without displacing other nodes after it. Asymptotically, the time complexity is $\Theta(1)$.

`removeLast()`: Removes the last node of the linked list and returns its reference to the calling program. A linked list must be traversed along its nodes (beginning from the first node) in order to reach the last node for removal. Asymptotically, the time complexity is $\Theta(n)$. It takes constant time if the pointer to the node before the last one is maintained. This could be done if a sequence of nodes were removed, one after another. Asymptotically, the time complexity is $\Theta(1)$.

`remove(int index)`: Removes the node at a particular position (given by the index) in the linked list and returns its reference to the calling program. A linked list must be traversed along its nodes (beginning from the first node) in order to reach the node for deletion at the given index. Asymptotically, the time complexity is $\Theta(n)$. It takes constant time if the pointer to the node before the node before the one that was deleted is maintained. This could be done if a sequence of nodes were removed, one after another. Asymptotically, the time complexity is $\Theta(1)$.

5 Doubly Linked List

A doubly linked list contains nodes with two pointers. One points to the next node and the other points to the previous node. These two pointers are called a forward pointer and a backward pointer. So, a doubly linked list can be traversed forward and backward. The java class for the node consists of three variables, an element (generic) object reference to data and two object references to the next neighboring nodes. Nodes can be non-contiguously stored in memory. The memory size due to the references to data and the next and previous nodes can be considered overhead. Asymptotically, the space complexity of memory overhead in a doubly linked list is $\Theta(n)$. Data access and reading can be faster than a linked list because traversal can be done in both directions. Asymptotically, the time complexity to access and read a data element in a linked list is $\Theta(n)$.

5.1 *Insertion and Deletion Operations for a Doubly Linked List*

For a doubly linked list, `addFirst` is always done in constant time no matter how big the doubly linked list, because displacement of the following nodes is not required. Asymptotically, the time complexity is $\Theta(1)$.

The `addLast` operation is always done in constant time no matter how big the doubly linked list, because displacement of preceding nodes is not required. Asymptotically, the time complexity is $\Theta(1)$.

For `add(index, E o)`, on average, half the doubly linked list must be traversed in order to reach the point of insertion. Asymptotically, the time complexity is $\Theta(n)$. Asymptotically, the time complexity is $\Theta(1)$, if a pointer to the last node inserted is maintained. This could be done if a sequence of nodes were inserted, one after another.

`removeFirst()`: Takes constant time no matter how big the doubly linked list, because displacement of the following nodes is not required. Asymptotically, the time complexity is $\Theta(1)$.

`removeLast()`: Takes constant time no matter how big the doubly linked list, because displacement of preceding nodes is not required. Asymptotically, the time complexity is $\Theta(1)$.

`remove(index)`: On average, half the doubly linked list must be traversed in order to reach the node for deletion at the given index. Asymptotically, the time complexity is $\Theta(n)$. Asymptotically, the time complexity is $\Theta(1)$, if a pointer to the node before the one that was deleted is maintained. This could be done if a sequence of nodes were removed, one after another.

6 Performance Evaluation

Performance evaluation of the add and remove operations for array list and linked list were undertaken on a 2018 MacBook Pro with 2.6 GHz 6-Core Intel Core i7 processor and 32 GB 2400 MHz DDR4 RAM. The profiling software and experiments were implemented using Java version 11.0.1 on Eclipse IDE version 4.10.0. The generic `LinkedList` class in the Java Collections framework is implemented as a doubly linked list. The time complexity of the `addLast` operation for both an array list and linked list as discussed in the previous sections is $\Theta(1)$. A time complexity of $\Theta(1)$ means that an operation takes constant time to complete irrespective of the data size. The time complexity does not provide information about the practical execution time of the operation. An operation that is $\Theta(1)$ might take 5 ms to complete when implemented one way and 50 ms to complete for a completely different implementation. Obviously, with respect to execution time, the one that takes 5 ms is the better choice of the two. This kind of information is important when a software developer needs to identify operations in software that perform poorly and improve upon their implementation.

6.1 Performance of Insertion Operations

As shown in Figs. 3 and 4, the practical performance (with respect to execution time and memory usage) of the addLast operation is different for array list and linked list implementations in the Java Collections framework. The memory profile was obtained using the Java Runtime class, and the software execution time was obtained using the Java System class. The profile of the heap memory allocated to the program is as follows: total memory is 512 MB and maximum memory is 8 GB. The objects of integer (wrapper class for the 4-byte *int* data type) were used to store data generated as uniform random integers in the range 0 to 1,000,000, where 0 is inclusive and 1,000,000 is exclusive. The objects are created on the heap. A logarithmic scale is used for *n* (the data size, i.e., the number of integers) on the x-axis of the graphs.

Figures 3 and 4 show how the execution time and memory size for calling addLast repeatedly (to create an array list or doubly linked list) increases as *n* increases. For a linked list, a separate node is created for each call of addLast which has greater memory overhead per data point and thus leads to greater total memory allocation compared to an array list. For $n > 200,000,000$, the memory usage for doubly linked list nears the maximum heap size (8 GB) and for array list nears 70% of the maximum heap size. The performance gap widens to the point where the performance for array list is 8.5 times faster than that for linked list. Also, the execution time and memory usage for array list are little more than doubles when the data size increases from $n = 100,000,000$ to $n > 200,000,000$. This is expected. For the same increase in data size, the memory usage for linked list is also little more than doubles; however the execution time is more than triples. This is because, in nearing the maximum heap size, there is greater overhead of growing the heap in the case of a linked list. Also, the Java garbage collector (GC) [6] runs more frequently,

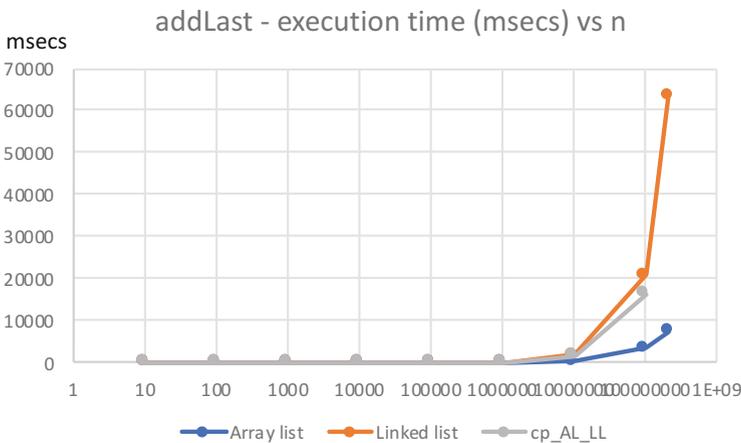


Fig. 3 Execution time vs data size for addLast and cp_AL_LL

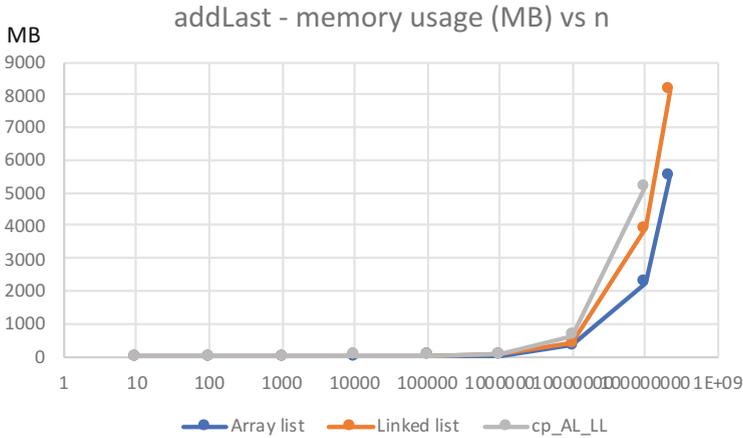


Fig. 4 Memory utilization vs data size for addLast and cp_AL_LL

when more than 70% of the heap is used [7]. GC is a program that deletes objects for which no object reference exists in the program. GC is run automatically and periodically by the Java virtual machine (JVM). It can be invoked using the Runtime class, but is non-deterministic, which means the exact start time of execution cannot be predicted.

The addFirst method shows similar performance to addLast for linked list, but worse performance for array list because of the overhead of right shifting each object reference one place. To construct a doubly linked list in the Java Collections framework that does not exceed the maximum data capacity of an array list, the author proposes that instead of using addFirst or addLast operations for linked list, the software developer should first create an array list and then use the addAll method to create the linked list from the array list. Figure 4 shows that this method (cp_AL_LL) uses more memory because both the array list and linked list are resident on the heap. This also means that the maximum possible data size of the linked list will be half of that which is possible with addLast. However, the proposed method performs better in constructing a linked list and takes 20% less time for $n = 100,000,000$. If greater data size is desired, then the heap memory size can be increased, which is possible in Eclipse or on the command line when running the program. Using a different data type such as Double (wrapper class for the 8-byte double data type) has negligible effect on the performance.

6.2 Performance of Deletion Operations

Figures 5 and 6 show how the execution time and memory size for calling removeLast repeatedly (to delete an array list or linked list) increases as n increases.

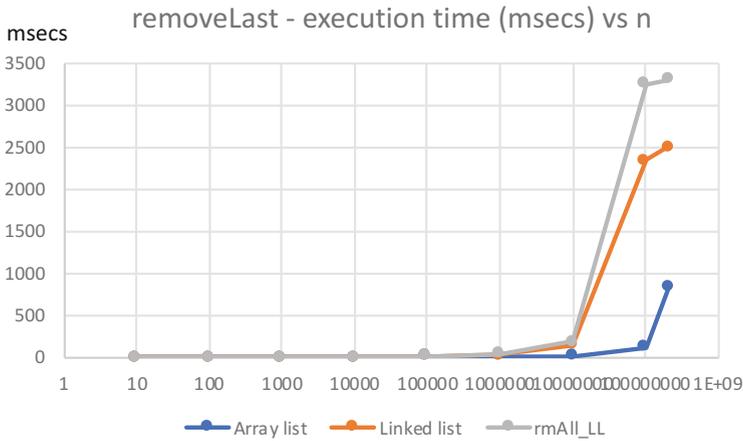


Fig. 5 Execution time vs data size for removeLast and rmAll_LL

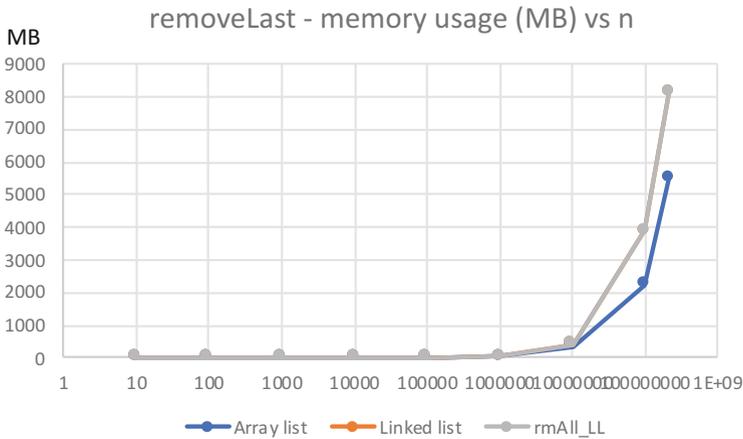


Fig. 6 Memory utilization vs data size for removeLast and rmAll_LL

In Fig. 6, the plot for linked list is hidden because it follows the same trajectory as the plot for the removeAll method for linked list. Also, the plots for the memory usage of linked list and array list in Fig. 6 follow the same trajectory as in Fig. 4. For $n > 200,000,000$, the performance gap widens to the point where the performance for array list is more than 2.5 times faster than that for linked list. This is because the GC runs more frequently when the heap is greater than 70% its maximum size. Also, prior to deletion of the linked list, the heap is already close to its maximum size; therefore there is no overhead to grow the heap, thus limiting the performance gap. Furthermore, Figs. 3 and 5 show that deleting a linked list is about 25 times faster than creating the link list and deleting an array list is about 10 times faster than creating the array list. It should be noted, however, that the GC did not run

during the repeated calls to `removeLast` and thus the execution time did not include the overheads for object deletion and reduction of the heap size.

As memory usage for linked list reduces to half the maximum heap size for $n = 100,000,000$, the performance gap widens to the point where the performance for array list is nearly 20 times faster than that for linked list, but the performance for linked list stays about the same. This almost static performance level for linked list even though the memory usage is cut by half is interesting and requires further investigation. The `removeFirst` method shows similar performance to `removeLast` for linked list, but worse performance for array list because of the overhead of left shifting each object reference one place. Also, as shown in Figs. 5 and 6, the `removeAll` method for the `LinkedList` class shows almost static performance level for linked list even though n and the memory usage are cut by half from nearly the maximum heap size for $n > 200,000,000$. Again, this is interesting and requires further investigation. Repeated calls to the `removeLast` method for deletion of the linked list performs about 25% faster than `removeAll` as the memory usage approaches maximum heap size for $n > 200,000,000$. Using `removeLast` also has the advantage of returning the objects containing the data points. The comparatively poorer performance of `removeAll` is due to its implementation which includes several recursive calls and conditional statements as fail safes. The `removeAll` method also calls `removeFirst`, and hence its performance for array list is extremely poor and reduces exponentially. If simply deleting all objects in the array list or linked list is desired, then the author proposes that the `clear` method be used. The `clear` method is implemented for the classes `ArrayList` and `LinkedList` and simply assigns a null reference to each data object reference. The `clear` method performs about 60% faster for array list and about 30% faster for linked list when compared to repeated calls of `removeLast` as the memory usage approaches maximum heap size for $n > 200,000,000$. The `clear` method also shows almost static performance level for linked list even though n and the memory usage are cut by half from nearly the maximum heap size for $n > 200,000,000$. This is interesting and requires further investigation.

7 Stack

A stack can be viewed as a special type of list, where the elements are accessed, inserted, and deleted only from the end, called the top, of the stack. Parsing algorithms used by compilers to determine whether a program is syntactically correct involve the use of stacks. Stacks can be used to evaluate arithmetic expressions. A stack is a last-in-first-out (LIFO) or first-in-last-out (FILO) data structure. It behaves like a stack of books. Objects are pushed (added) to the stack on top of previous objects. Objects are popped (removed) from the top of the stack. Other important applications of stacks are in recursive backtracking and method calls in computer programs.

A linked list can be used to implement a stack because insertion and deletion operations are efficient when done at the front end of the linked list. However, memory overhead is greater than that of an array list. For a doubly linked list, insertion and deletion operations are efficient irrespective of being done at the front end or back end. A doubly linked list supports search better than a linked list, if search functionality is desired. However, memory overhead is greater than a linked list.

The best choice for implementing a stack of objects is an array list because it is faster to add and remove objects, has less memory overhead than a linked list and search, and performs much better than a doubly linked list due to random access. However, for data that consists only of numbers, an array of primitive data type (the type depends on the range of values in the data) has the least memory overhead and best search performance. However, an array is not dynamically resizable and additional implementation will be required to implement a stack that is not of fixed capacity. Furthermore, on systems that allow heap memory size expansion to accommodate extremely large data sizes, the maximum capacity ($2^{31}-1$) of arrays and array lists may prove restrictive. Also, the size variable for the `LinkedList` class is of type `int`, thus restricting the maximum positive value to $(2^{31}-1)$. In this case, a doubly linked list should be implemented, because its capacity is restricted only by the maximum heap size. The size variable could be implemented as type `long` (8-bytes) for a maximum positive value of $(2^{63}-1)$. Alternatively, the `BigDecimal` class could be used to store and manipulate extremely large integers as `Strings`.

8 Queue

A queue represents a waiting list. A queue can be viewed as a special type of list, where the elements are inserted into the end (tail) of the queue and are accessed and deleted from the beginning (head) of the queue. Queues are widely used in modeling and simulations. They are used in serving requests of a single shared resource (printer, disk, CPU), transferring data asynchronously (data not necessarily received at same rate as sent) between two processes (IO buffers), and interrupt handling in operating systems. A queue is a first-in-first-out (FIFO) data structure. It has the same methods as a stack except that the method `push` is replaced by `enqueue` and the method `pop` is replaced by `dequeue`. The method `enqueue` adds an object to the end of the queue and `dequeue` removes an object from the front of the queue.

A priority queue can be perceived as a special type of queue where data is prioritized for deletion. The data at highest priority is deleted first.

The best choice for implementing a queue is either a linked list or a doubly linked list. Removal operations from the front of a linked list and doubly linked list are efficient. If less memory overhead is desired, then the best choice is a linked list. However, if increased search capability is desired, then the best choice is a doubly linked list. This is because a doubly linked list can be traversed from both directions.

For an array and arraylist, removals from the front end are inefficient because all the following elements must be moved one position toward the front end.

9 Conclusion and Future Work

Insertion and deletion operations of data structures that are asymptotically identical might display severe performance gaps practically. Some of these gaps are identified in this paper and alternative approaches leading to improved performance are proposed. As per the performance evaluation, it was found that a stack can be best implemented using an array list and a queue can be best implemented using a linked list. The stack class in the Java Collections framework uses the class Vector which gives similar performance to an array list. However, Vector is deprecated and class ArrayList is essentially its replacement. Furthermore, to implement a queue the ArrayDeque class can be used because it implements a circular array in which the left shift of elements is eliminated for the removeFirst method. Furthermore, searches are faster because of random access in ArrayDeque. The average case, where data is inserted or deleted from the middle of the array list or linked list, takes about the same time to execute for a single operation. This is expected as right or left shifts are required for an array list and traversal of preceding objects are required for a linked list. Furthermore, the locality of reference for a linked list is far poorer than that of an array list causing greater paging overhead with respect to the CPU cache.

Future work can involve the evaluation of multiple successive calls and maintenance of a reference to avoid multiple traversals in a linked list. Using the Iterator class can speed up the above operation because it maintains references to the nodes in the linked list. Also, for future work, compiler optimization effects on performance can be evaluated. The approach adopted in this paper can be leveraged to evaluate practical performance of data structures implemented in other programming languages such as the C++ standard template library and compare and contrast them with the Java Collections framework.

References

1. Oracle JavaSE Documentation, The collections framework (2018), <https://docs.oracle.com/javase/7/docs/technotes/guides/collections/index.html>. Accessed 23 June 2020
2. Wikipedia The Free Encyclopedia, Data structure (2020), https://en.wikipedia.org/wiki/Data_structure. Accessed 23 June 2020
3. Wikipedia The Free Encyclopedia, Linked list (2020), https://en.wikipedia.org/wiki/Linked_list#Linked_lists_vs._dynamic_arrays. Accessed 23 June 2020
4. C.A. Shaffer, *A Practical Introduction to Data Structures and Algorithm Analysis*, Second edn. (Prentice Hall, New Jersey, 2001)

5. Y.D. Liang, *Introduction to Java Programming and Data Structures, Comprehensive Version*, Eleventh edn. (Pearson, New York, 2017)
6. Oracle Learning Library, Java Garbage Collection Basics (2012), <https://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>. Accessed 23 June 2020
7. IBM Knowledge Center, Heap Sizing Problems (2020), https://www.ibm.com/support/knowledgecenter/SSYKE2_8.0.0/com.ibm.java.vm.80.doc/docs/mm_heapsize_problems.html. Accessed 26 June 2020

Part V
Software Engineering, Dependability,
Optimization, Testing, and Requirement
Engineering

Securing a Dependability Improvement Mechanism for Cyber-Physical Systems



Gilbert Regan, Fergal Mc Caffery, Pangkaj Chandra Paul, Ioannis Sorokos, Jan Reich, Eric Armengaud, and Marc Zeller

1 Introduction

Cyber-physical systems (CPS) harbor the potential for vast economic and societal impact in domains such as mobility, home automation, and delivery of health. At the same time, if such systems fail, they may harm people and lead to temporary collapse of important infrastructures with catastrophic results for industry and society [1]. There are two core challenges while assessing the dependability of a CPS. First, the inherent complexity of modern CPS [2] and the resulting complex market organization requiring the tight cooperation between different teams, expertise, and institutions, while managing confidentiality issues. The second challenge is related to the increase of connectivity, e.g., through machine-to-machine cooperation enabled by the Internet of things, which introduces a new dynamic in system operation [2]. As a result, cyber-physical systems of systems (CPSoS) come together as temporary configurations of CPS, which dissolve and give place to other configurations. This leads to a potentially infinite number of

G. Regan (✉) · F. M. Caffery · P. C. Paul

Lero @DKIT, Dundalk, Ireland

e-mail: Gilbert.regan@dkit.ie; fergal.mccaffery@dkit.ie; pangkajchandra.paul@dkit.ie

I. Sorokos · J. Reich

Fraunhofer IESE, Kaiserslautern, Germany

e-mail: ioannis.sorokos@iese.fraunhofer.de; jan.reich@iese.fraunhofer.de

E. Armengaud

AVL List GmbH, Austria, Turkey

e-mail: eric.armengaud@avl.com

M. Zeller

Siemens, Munich, Germany

e-mail: marc.zeller@siemens.com

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_35

511

variants, with cooperation between systems potentially not analyzed during design time.

The DEIS project [3] addresses these important and unsolved challenges by developing technologies that form a science of dependable system integration. In the core of these technologies lies the concept of a Digital Dependability Identity (DDI) of a component or system. The DDI targets (1) improving the efficiency of generating consistent dependability argumentation over the supply chain during design time and (2) laying the foundation for runtime certification of ad hoc networks of embedded systems.

Contribution of this paper is to present the protocol for securing the DDI while it is in transit and at rest. The paper is organized as follows: Sect. 2 presents an overview of the DDI, while the research methodology is presented in Sect. 3. Section 4 presents the protocol for securing the DDI while it is in transit, while Sect. 5 presents the protocol for securing the DDI while it is at rest. Finally, Sect. 6 presents validation results, while Sect. 7 concludes this work.

2 Overview of DDI

Assurance cases represent the backbone of modern dependability assurance processes, because they record the dependability requirements to be fulfilled by a system (of system) in an intended operational environment together with the evidences that support the requirement's validity in the finally implemented system. All produced dependability engineering artifacts using such evidence are motivated by an uncertainty about whether a dependability claim about the system is actually fulfilled.

Since there is an interrelation between the system, its dependability claims, and the supporting evidence artifacts that exist in the real world, we claim this should also be the case for the system's model-based safety reflection, i.e., its DDI (see Fig. 1).

DDIs represent an integrated set of dependability data models that may be (semi-) automatically analyzed, generated, or manipulated during the execution of safety engineering processes. A DDI contains information that uniquely describes all the dependability characteristics of a system required for certifying the system's dependability. DDIs are formed as modular assurance cases, and their composability allows for the (semi-)automatically synthesizing of system DDIs from the DDIs of the subcomponents. The DDI of a system contains (a) claims about the dependability guarantees given by a system to other systems and derived system dependability requirements and (b) supporting evidence for those claims in the form of various models and analyses. For security assurance, it contains a threat and risk analyses (TARA) which is composed of attack trees, while for safety assurance, hazard and risk analyses (HARA), architecture modeling, and failure propagation modeling such as fault trees, FMEA, or Markov chains are supported.

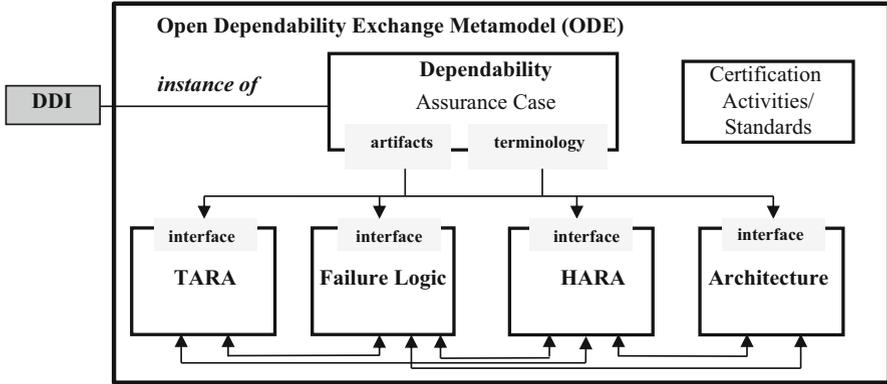


Fig. 1 The Open Dependability Exchange Metamodel (ODE)

Due to the integration and standardization of these models in the Open Dependability Exchange Metamodel (ODE), a self-contained system dependability package can support many dependability engineering activities of the system life cycle. A video of the DDI being employed in a truck platooning use case can be seen here [4].

3 Methodology

The data confidentiality, integrity, and availability (CIA) triad is a common concept to ensure data security. The attacker can launch various attacks to compromise the CIA of data in transit and at rest. To mitigate these attacks and to assure the CIA of the DDI, this research designed a security protocol which is presented in Fig. 2. The proposed security protocol consists of three key stages: (1) identify the possible threats, (2) identify the security controls, and (3) evaluate the security of DDI.

Risk assessment process such as NIST 800-30 [4], CIS RAM [5], or a threat modeling technique such as STRIDE [6] can be used to identify the possible threats. As the focus of this research is on securing the DDI and not the whole application, attacks such as denial of service (DOS), eavesdropping, and data modification are considered.

The next stage in the protocol is to identify mitigating security controls. There are several standards, guidelines, and frameworks which provide the security controls to put countermeasure against various attacks. Examples include ISO/IEC 27002 [7], NIST 800-53 [8], and the NIST Cybersecurity Framework [9]. This research adopted ISO 27002 as a source for selecting security controls because it is a widely used data security standard. This standard provides a large list of security controls with very high-level implementation details. Exclusion criteria, as detailed in Fig. 2, were used to select the appropriate security controls for assuring the security of the

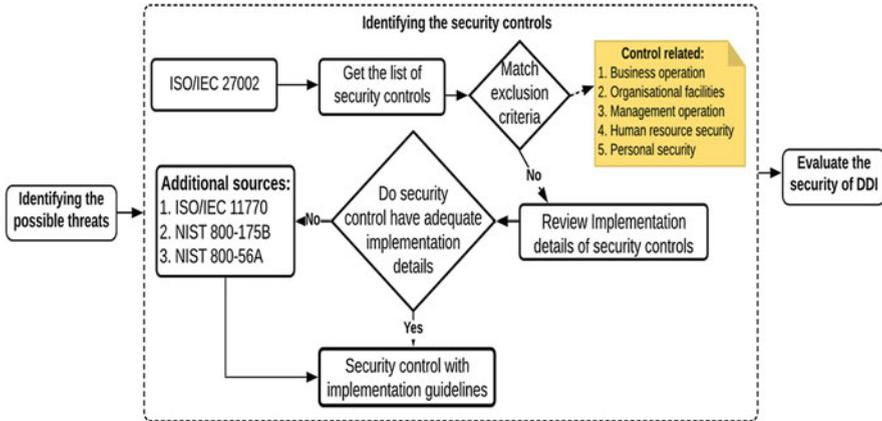


Fig. 2 DDI security protocol

DDI in transit and at rest. For example, controls related to business and management operation, or to personal security, were excluded. This resulted in the following four key security control categories: access control, cryptography, physical and environmental security, and communications security. With the appropriate security controls selected, the next step was to review the implementation details of each selected control. If any of the selected security controls did not have adequate implementation details, then sources external to the ISO/IEC 27002 standard were employed. For example, ISO/IEC 27002 proposes to use cryptography to assure the integrity and confidentiality of data. However, this standard does not provide enough detail for implementing cryptography in an application. Therefore, external sources such as the NIST 800-175B Cryptographic Standard [10] and the ISO/IEC 11770 key management standard [11] were reviewed for implementation detail.

To evaluate the proposed security protocol, a truck platooning use case was selected which was implemented via a simulator framework. The simulation involves two trucks, with their intercommunication implemented via the Robot Operating System (ROS) [12]. By default, messages published to a ROS topic are in plaintext. We considered the following three types of attacks: First, an attacker could attack a critical service which monitors sensor variables and force it to be deactivated. We deemed this to be an attack on the service’s availability. Second, attackers could eavesdrop on the packet traffic and capture the plaintext ROS messages which would violate the confidentiality of the platoon’s information. Finally, attackers could alter the content of the exchanged ROS messages and insert incorrect or misleading information to compromise the system’s integrity.

4 Securing the DDI in Transit

DDI data can be in transit between components within a system, or between system to cloud server, or between systems.

4.1 DDI in Transit Between System Components

Communication between components in a system can be hardwired or wireless. Whether the connection is hardwired or wireless, the following measures are required:

Communicating components of CPS can be known (i.e., pre-certified) or unknown. If an entity is known, a pre-signed key can be used to secure communication. If an entity is unknown, the widely used Elliptic-Curve Diffie-Hellman (ECDH) protocol can be used to secure communication. This protocol is standardized by NIST in SP 800-56A and allows two parties to establish a shared secret over an insecure channel. Basic Diffie-Hellman (instead of ECDH) can be used as well, as it has lower memory and power requirements; however, ECDH produces a stronger secret key as ECCH uses algebraic curves method to generate the key.

Additionally, policies for firmware upgrade and installation and policy for port management auditing are required. These policies assist with ensuring the confidentiality and integrity of data in transit between system components.

4.2 DDI in Transit from System to Cloud Server

For scenarios where DDI packages are transmitted between CPS and cloud services, it is recommended to use the HTTPS (Hypertext Transfer Protocol Secure) protocol. HTTPS establishes an encrypted link using Secure Socket Layer (SSL) or Transport Layer Security (TLS). TLS is the new version of SSL. TLS establishes an encrypted link using a TLS certificate which is also known as a digital certificate. TLS can be configured to ensure the following properties:

- Private connection via symmetric cryptography
- Authentication via public key cryptography
- Data integrity via a message authentication code (MAC)

4.3 DDI in Transit from System to System

Individual constituent systems can be unknown, known to each other or known centrally by some management authority. Unknown parties are inherently untrusted and are potential avenues for malicious attacks on confidentiality and integrity. Therefore, where communication involves directly or indirectly untrusted parties, exchanged data must be secured at the system boundaries. To secure the transit of data between systems, the following considerations need to be taken into account:

- Are the systems involved in the exchange pre-certified or do they need to be certified on the fly? For pre-certified systems, each CPS' key will be stored in the Key Management Service (KMS) and can be shared from there. For systems that need to be certified on the fly, each system can generate their own encryption key and share it. Additionally, if a system is known centrally, it can retrieve the key from the KMS and share from there.
- Choice of encryption key, i.e., asymmetric or symmetric? The choice of cryptography technique (asymmetric/symmetric) depends on device resources (i.e., computational power, memory, etc.) and the KMS cost. In general, symmetric encryption requires less device resources, is less costly, and requires minimal effort for key management.
- Is the communication to be one-to-one between systems, and/or is the message to be broadcast to all systems in the network?

The following section portrays how the system-to-system protocol can be applied to the platoon use case.

4.4 System-to-System Protocol Applied to Platoon Use Case

Figure 3 provides the communication model for the platoon use case. Communication can be “bidirectional” or “bidirectional with centralized broadcast.”

The options for securing both of these communication models for pre-certified and certified on-the-fly systems are now provided.

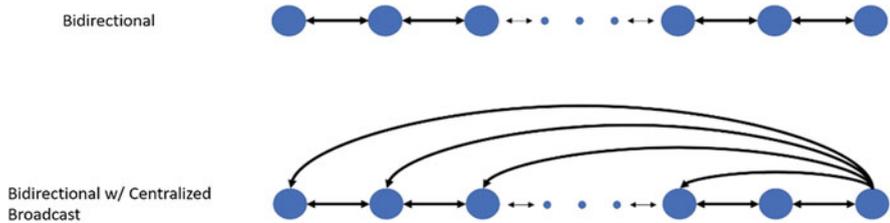


Fig. 3 Platoon communication models

Bidirectional with Pre-certification

With pre-certification, there are two encryption options available – using a symmetric or an asymmetric key. The choice of asymmetric versus symmetric encryption is decided based on computational cost and key management cost. The choice of encryption approach ultimately depends on trade-off analysis during development.

Asymmetric encryption is generally more resource intensive. For asymmetric encryption, the central authority always maintains a private key, and each system has a public key. There are two ways to generate asymmetric keys:

- RSA – for a shared public key
- X.509 – for each system having their own individual public key

Symmetric key encryption means each system shares their symmetric keys with the system they want to communicate with.

For the platoon use case with *asymmetric key encryption*, RSA with shared public key option is chosen because a certificate authority is required for managing X.509 certificate, and deploying such a service is costly and time-consuming. Using *symmetric key encryption*, the symmetric key will be generated using AES 256 which is a widely recognized standard. Each truck shares their symmetric keys with the truck they want to communicate with.

With pre-certified key sharing, all keys are generated and stored centrally in the Key Management Service (KMS). Thus, each authenticated truck can obtain both their own key, as well as their neighbor's truck key from the KMS.

Bidirectional with On-the-Fly Certification

Asymmetric key encryption – Public Key Infrastructure (PKI) certification (e.g., via X.509) cannot be used because a certificate authority is needed. Deploying such a service is too costly and time-consuming for an ad hoc network of systems. For RSA certification, a public and private key pair must be generated, between two systems.

There are many ways to exchange asymmetric keys; the following are reasonable options:

- A master key or signature (manufacturer-specific) is used to encrypt the keys and share them over the communication channel.
- Keys are shared during ACK handshake with Message Authentication Code (MAC)

For the platoon use case, using asymmetric key encryption, RSA certification with a public and private key pair is chosen because it is cheaper and less time-consuming when compared to X.509.

- When two trucks want to communicate and share keys in this platoon scenario, there are two options which they can adopt. The choice of which option is determined by the manufacturer during development.
- Option 1: Truck 1 will use a master key or signature (manufacturer-specific) to encrypt the keys and share with the follower truck over the communication channel.

- Option 2: The two trucks can share keys during ACK handshake with Message Authentication Code (MAC).

For the platoon use case using *symmetric key encryption*, each truck must be approved by a central authority (CA) server, and this server will generate symmetric key using AES 256 for each truck. The CA notifies each truck of its neighbors (if any) and shares the neighbors' keys. At this point, each truck knows the key for its preceding and following trucks, so their bidirectional communication can be secured.

Bidirectional with Broadcast Message with Pre-certification

In this model, trucks communicate with their neighbors, but the lead truck can also broadcast to each truck. For broadcast messages from the lead truck, asymmetric encryption can be used by sharing the manufacturer's specific public key with all members in the platoon. If symmetric encryption is preferred, a known common key must be shared with all platoon members. The techniques for exchanging keys and the use of a KMS, as described in the "Bidirectional with Pre-certification" section, can be applied again.

Bidirectional with Broadcast Message: On-the-Fly Certification

Asymmetric Encryption – For one-to-one communication between two platoon members, the same recommendation as per the "Bidirectional with On-the-Fly Certification" section can be used, i.e., using RSA to generate public-private keys between platoon member pair.

For message broadcasting, the leader's public key will be shared with all members in the platoon. The same recommendations for sharing keys as per the "Bidirectional with On-the-Fly Certification" still apply.

Symmetric Encryption – Assumption: Any truck wanting to join platoon is approved by platoon leader. The leader will generate each truck's symmetric key using AES 256 and exchange using the Diffie-Hellman technique. The recommendations for platoon member communication are the same as per the "Bidirectional with On-the-Fly Certification" section.

For broadcast message, the leader will generate a generic key and share it with each member of the platoon separately. To share generic key securely, the leader will encrypt the generic key using each individual's key (which was generated during one to one). This encrypted generic key can now be decrypted by each individual member.

5 Securing the DDI at Rest

The DDI at rest is the case where the DDI is stored statically within a CPS, for instance, in local memory or on a cloud service. When the stored data involves intellectual property concerns, or is significant for the system's functionality, or is personal data, then it may be prudent or even mandatory to encrypt the data.

Asymmetric and symmetric encryptions are two mutually exclusive options. Asymmetric keys (also known as public keys) require high computational power for encryption and decryption but are considered very secure. Symmetric keys are comparatively cheaper, require less computational power, and introduce less communication delay. For the above reasons, symmetric keys are recommended by default for DDI applications.

There are two further options to consider with symmetric keys:

- Stream ciphers encrypt and decrypt data one bit at a time which means that they are particularly well-suited to real-time hardware-based applications, such as audio and video applications. Stream ciphers are weaker and less efficient than block ciphers when it comes to software applications and are less frequently used in that sphere. The encryption key size is often the same length in both approaches;
- For block ciphers, strong algorithms mean that reverse engineering the cipher, or determining which functions were performed on each block, or their order, is virtually impossible.

For symmetric cryptographic encryption, the Advanced Encryption Standard (AES) 256 is recommended. AES 256 is a widely recognized symmetric key and recognized by standards bodies, i.e., ISO 18033-3 (Security Techniques Standard) and NIST 800-175B (Using Cryptographic Standards in the Federal Government). Symmetric key block cipher algorithms include SEED (block), Camellia (block), CAST-128 (block), Blowfish (block), AES (block), and DES (block).

Encryption Key Storage

After the encryption keys have been generated, consideration needs to be given as to how they will be stored in the system and in the cloud. For cloud storage, the KMS can be used. A cloud service with FIPS 140-2 (Cryptographic Modules Standards), which use hardware security modules (HSMs) to generate and protect keys, should be chosen. HSMs are considered more secure than software encryption for generating encryption keys. For system storage, the key can be stored in the erasable programmable read-only memory (EPROM).

Cloud Service-Specific Security Measures

To secure data in the cloud, an “Encryption at Rest” feature should be enabled. This means the hard drive in the cloud is encrypted. Additionally, for port management, ensure that only those ports that you require are open. Finally, every cloud has an Identity Management Service, which needs to be configured to ensure appropriate Identity Management for Access Control (IMAC).

DDI File Security in the Cloud

For files stored in cloud “Storage Service,” the storage should be encrypted at file level. Additionally, files should not be publicly accessible, i.e., only authorized access by application.

Application Security

A web application firewall (WAF) should be employed in the cloud. The WAF helps protect against attacks such as DDOS, SQL injection, path traversal, etc. The firewall will help ensure the availability of the system.

Database Security

Appropriate access control and role management should be employed so that only the application has access to the database. To encrypt the hard drive of the database, it should be configured with “Encryption at Rest” service. Finally, all sensitive information (Personal Identifiable Information) in the database should be encrypted using field-level encryption, e.g., AES 256.

6 Results

To demonstrate the attack in our use case, we used the built-in ROS command-line utility “rostopic echo” that directly outputted the contents of the messages exchanged by the vehicle systems. The result can be seen on the left of Fig. 4. After enabling encryption and message authentication, message contents instead only include the encrypted payload and the MAC, seen on the right side of Fig. 4.

To address availability, we implemented a service supervisor for the sensor monitor. The attack defended against would attempt to take down the monitor service, debilitating the vehicle. Such attacks are possible in ROS via the RosPenTo [13]. After taking down the sensor monitor, the following vehicle no longer reacts to changing conditions, leading to potentially unsafe driving behavior. The service supervisor is activatable via the simulation user interface, seen in Fig. 5. Once enabled, taking down the service leads to the supervisor detecting and re-launching the monitor, thereby ensuring availability and safe driving behavior.



```
wlittenal@49f490ebeb:~/catkin_ws$ rostopic echo /carla/follower_vehicle/vehicle_status
velocity: 15.3566570282
acceleration: 0.631387412548
orientation:
  pitch: 0.15477523804
  yaw: 179.970668932
  roll: 0.00103819917422
control:
  header:
    seq: 0
    stamp:
      secs: 0
      nsecs: 0
    frame_id: ''
  throttle: 0.399438215359
  steer: -0.8009144604884554
  brake: 0.0
  hand_brake: False
  reverse: False
  gear: 0
  manual_gear_shift: False
...

wlittenal@49f490ebeb:~/catkin_ws$ rostopic echo /SystemState
mac: "22fb574beff24e013d711fa3dfc4e61c349bb672858c0b49255c16f9429e4"
content: "0b6f7bb7d016113a66142d1647738e3031c0428e56d89ed49d306d44708ede69"
...
```

Fig. 4 Encryption and MAC application

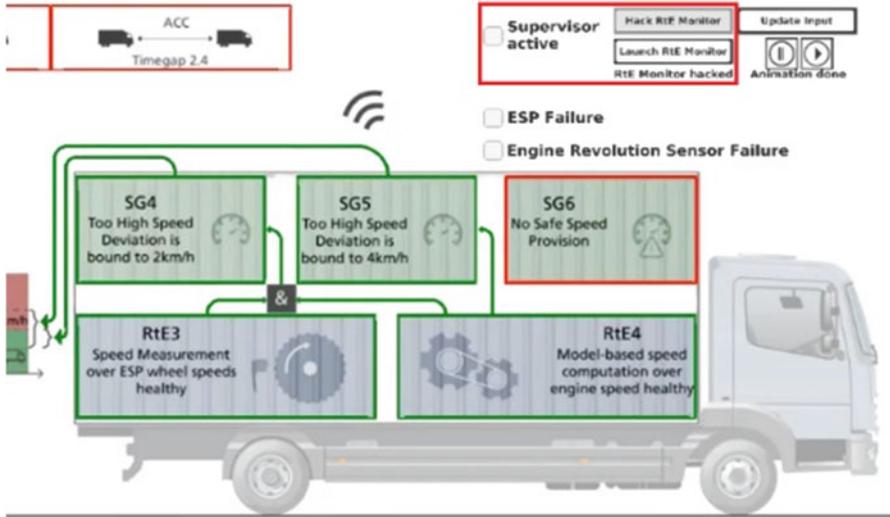


Fig. 5 Partial view of simulator interface

7 Conclusion

Securing open and adaptive CPS is paramount to maintaining their effectiveness and delivering their full potential to users and infrastructure. The DEIS project developed the concept of the DDI to support generic information exchange across CPS. In this publication, we presented our investigation into recommended security protocols that aim to provide base coverage across a diverse set of CPS application scenarios, e.g., securing the DDI in transit, at rest, and more. These preliminary recommendations are by no means exhaustive, and it will be useful to expand upon them in future work.

As part of DEIS, we chose a subset of the recommended protocols to implement and evaluate within a truck platooning use case. We identified attacks that covered the standard confidentiality, integrity, and availability properties of the platoon CPS. We then implemented our recommended protocols successfully against the chosen attacks. The above use case should provide a reasonable basis for security analysis and protection for applications of a similar nature to the platoon system investigated. For more diverse CPS applications, the recommended protocols presented earlier can be reviewed and adjusted to secure DDI and/or similar exchanged information concepts.

Moving forward, we will be investigating in more detail means of incorporating privacy-specific threat analyses and protection methods, such as LINDDUN [14].

Acknowledgments This paper is supported by the European Union’s Horizon 2020 research and innovation program under grant agreement no. 732242. It is also supported in part by Science Foundation Ireland grant 13/RC/2094.

References

1. R. Wei, T. Kelly, R. Hawkins, E. Armengaud, DEIS: Dependability engineering innovation for cyber physical systems. Lecture Notes in Computer Science, vol. 10748 (Springer, Cham, 2018)
2. Platforms 4CPS. [Online]. Available: https://www.platforms4cps.eu/fileadmin/user_upload/Deliverable_1.2_European_ecosystem_and_market_opportunities_assessment.pdf. Accessed: 21 Mar 2020
3. DEIS. [Online]. Available: <http://deis-project.eu/>. Accessed: 21 Mar 2020
4. DEIS Truck platooning. Available: <https://www.youtube.com/watch?v=Vdn-TCGxzgA>
5. NIST Special Publication (SP) 800-30, Rev 1, Guide for Conducting Risk Assessments. Available: <https://www.nist.gov/privacy-framework/nist-sp-800-30>
6. Center for Internet Security. Risk assessment method. Available: <https://learn.cisecurity.org/cis-ram>. Accessed: 21 Mar 2020
7. Software Engineering Institute. Threat modelling: 12 available methods. Available: https://insights.sei.cmu.edu/sei_blog/2018/12/threat-modeling-12-available-methods.html. Accessed: 21 Mar 2020
8. ISO/IEC 27002:2013 — Information technology — Security techniques — Code of practice for information security controls
9. NIST SP 800-53 Rev. 4 Security and Privacy Controls for Federal Information Systems and Organizations. Available: <https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final>
10. NIST Cybersecurity Framework. Available: <https://www.nist.gov/cyberframework>
11. NIST SP 800-175B Guideline for Using Cryptographic Standards in the Federal Government: Cryptographic Mechanisms. Available: <https://csrc.nist.gov/publications/detail/sp/800-175b/final>
12. ISO/IEC 11770-1:2010 [ISO/IEC 11770-1:2010] Information technology — Security techniques — Key management — Part 1: Framework
13. Robot Operating System. Available: <https://www.ros.org/>
14. B. Dieber, R. White, S. Taurer, B. Breiling, G. Caiazza, H. Christensen, A. Cortesi, *Robot Operating Systems (ROS)*, Studies in Computational Intelligence, vol 831 (Springer, Cham, 2019)

A Preliminary Study of Transactive Memory System and Shared Temporal Cognition in the Collaborative Software Process Tailoring



Pei-Chi Chen, Jung-Chieh Lee, and Chung-Yang Chen

1 Introduction

Contemporary software development is dynamic in nature [14]. Because of this nature, software development project teams often need to alter their process. Such an activity of customizing standard processes to meet the project's needs is called software process tailoring (SPT) [2, 35, 40, 41]. Because SPT critically determines how a software project is conducted, its performance merits an investigation [15]. From a team management perspective, software teams play a core role and are primarily responsible for SPT execution. However, in the SPT context, the association between teams' operations and behaviors and SPT performance remains unknown. Accordingly, we propose a study to comprehend teams' operational mechanism in achieving higher levels of SPT performance.

The essence of SPT is knowledge- and learning-intensive. It requires team members, as task owners with diversified domain knowledge, to integrate, collaborate, and mutually learn to make suitable tailoring decisions regarding project tasks. Thus, in such an attempt, we consider the team's transactive memory system (TMS) that may play a critical role. Theoretically, TMS refers to shared understanding of where specialized knowledge exists in the team, thereby effectively integrating team members' distributed and complementary expertise and optimizing the value

P.-C. Chen
Trend Micro Inc., Irving, TX, USA

J.-C. Lee
International Business Faculty, Beijing Normal University Zhuhai, Zhuhai City, China

C.-Y. Chen (✉)
Department of Information Management, National Central University, Taoyuan City, Taiwan
e-mail: cychen@mgt.ncu.edu.tw

of members' knowledge [1, 4, 16, 31]. In this regard, the precise role of the TMS on software teams should be investigated. Hence, this study explores how a software team's TMS can improve SPT performance, in terms of efficiency and effectiveness of SPT [15, 41], given the characteristics of SPT.

As mentioned above, SPT is a collaborative activity that requires members to reciprocally coordinate their distinct tailoring missions, tasks, and specialized knowledge. However, performing SPT may be conflictual. Specifically, in discussing the re-planning of the development, team members often have different temporal perspectives and propensities [21, 22], which may trigger temporal conflicts [29, 34]. During SPT, such conflicts are amplified because the process involves disputes among members about time-related issues, such as different perceptions of schedules and deadlines, different pacing styles to accomplish a tailoring activity, and members' different views regarding the length of task durations. Temporal conflicts may decrease a software team's synchronization and coordination [9] in performing SPT. Thus, we focus on STC to explore how it exerts a contextual effect in tailoring results and outcomes.

In this context, three research questions are considered: (1) Does a software team's TMS contribute to SPT performance, and if so, how? (2) Does task conflict moderate the effect of TMS on SPT performance, and if so, how? (3) How does the moderating role of STC influence the relationship of TMS-SPT performance? To answer these questions, our entire study consists of two parts: (1) conceptualization of a research model and (2) empirical investigation of the research model. This paper serves as the first part with the goal to review the theoretical background for developing the research model. The remainder of this study is organized as follows. Sections 2, 3, and 4 review the theoretical backgrounds of this study. Section 5 presents the theorized model and the propositions. Section 6 concludes and outlines future research directions.

2 Software Process Tailoring (SPT)

The nature of software development is dynamic and fickle. Software projects have unique characteristics, such as the diverse requirements of customers, the technical complexity, and different sizes and scopes of projects. Therefore, no single software process can be fully applied to all projects [32, 41]. In this sense, software processes need to be tailored to accommodate a particular project's requirements and environments ([26, 40]). Specifically, SPT includes four steps [41]. The first step refers to the assessment of project goals and environments, which means that a software project team may need to adjust the project environment (e.g., budget, schedule, personnel) and project goals (e.g., the purpose of the software product and the derived quality requirements) to ensure consistency between the two.

The second step is to assess the resulting impacts due to tailoring. These impacts include corresponding changes or bad fixes on other processes to be modified and

the resources, budget, and personnel to be rearranged for the modified processes. The third step refers to identifying strategies of process modification to mitigate the impact of change. Tailoring strategies include, but are not limited to, the decision to expand, delete, replace, or simplify process elements. These strategies are used to alter the form, frequency, granularity, and scope of process elements to make the process suitable for a specific project environment [10]. The final step includes validation and evaluation of the process tailoring. Specifically, this step refers to the assessment of the influence on the project outcome after the implementation of the tailoring decision [25].

Although software development is collaborative in nature, conducting SPT is a challenging process. These challenges are twofold. First, in addition to the distinct characteristics of team members, conflicts primarily stem from their interdependent work relationship in development [18, 30]. Such conflicts are addressed in a later section. Second, the SPT is a knowledge-intensive activity that requires the team to possess and handle a great deal of product, process, or project knowledge. Product knowledge is knowledge regarding the product or software features and how they relate to other products, standards, and protocols. Project knowledge refers to knowledge about resources, deliverables, timing, milestones, increments, and quality targets. Process knowledge denotes knowledge about business processes, workflows, responsibilities, supporting technologies, and interfaces between processes [5].

3 Transactive Memory Systems (TMS)

Theoretically, TMS is a shared system that people in relationships develop to encode, store, and retrieve information about different substantive domains [28]. TMS can be regarded as a collaborative division of labor to allow a team to learn, remember, and communicate from complementary knowledge domains. Through TMS, team members can perceive who knows what and how [23, 28]. Team members can obtain other's knowledge and know-how by establishing a positive social interaction process [11]. Moreover, TMS facilitates a cross understanding of teammates' distributed knowledge, which can help in better coordinating and applying the knowledge to tasks [11, 17].

In the SPT context, conducting tailoring tasks involves the redesign of various and interdependent professional tasks and procedures. Thus, it requires intensive knowledge exchange and transfer among team members. In the literature, existing TMS studies mostly emphasize general teams' TMS (e.g., [3, 11, 27, 38]). In software development environments, however, the composition of the software project team is dynamic and temporary. Such a team setting, in addition to the aforementioned conflictual nature of the SPT process, may lead to discrepancies between the nature of SPT and the harmonious development of TMS. In this situation, it remains unknown whether a software team's TMS exerts an effect on

SPT tasks. Therefore, this study attempts to investigate and examine how TMS influences the performance of SPT.

4 Shared Temporal Cognitions (STC)

In theory, STC refers to a shared understanding among team members regarding the time-related aspects of a collective task execution, such as meeting the deadline, (sub)task completion times, and the pacing style of task implementation [22, 29]. STC helps members to have a similar attitude, orientation, and perspective on time and to foresee and comprehend each other's actions and thus to adopt more compatible work patterns. This may reduce the temporal diversities and differences among members, contributing to higher levels of temporal synchronization and enhancing the harmony and coordination of group task activities [29, 37]. Scholars have shown that STC helps streamline team processes, which in turn benefits team outcomes, and it decreases the extent of temporal and process conflicts, thereby increasing team performance [8, 22, 29].

As mentioned earlier, SPT is a conflicting process. Moreover, the heterogeneity of members also imposes temporal diversities and differences on a team [21, 22]. Because of the divergent temporal personalities of team members, future tasks or missions at the same time distance result in members' different ideas and perceptions, which in turn lead to different behaviors and decisions [19, 36]. For example, the duration of the residual half-month of a software project can be expressed as "the remaining half of a month" as a positive temporal frame or "still having a half month" as a negative temporal frame [24]. In this case, with the same time limit, individuals will have different perceptions, orientations, and cognitions due to different temporal frames [12]. These different temporal construal effects may lead to temporal conflicts among members [9, 29, 34] when performing tailoring tasks. Given that temporal conflicts that may exist in teams and because TMS relies on a well-coordinated team environment, the way that STC are contextually exerted on the mechanism of a team's TMS for an effective and efficient SPT decision is unknown. Therefore, this study explores the moderating effect of STC on TMS-SPT performance.

5 Development of a Theoretical Model

The development of a software project involves various types of knowledge and communication in different professional fields, such as engineering, administration, design, process assurance, and other management expertise (e.g., legal, customer relations) [20]. This complex and dynamic interaction and knowledge exchange may be amplified when conducting SPT that is conflicting in nature. Performing SPT requires the team to possess and handle a great deal of knowledge (i.e., product,

process, or project knowledge). From the theory of bounded rationality and the perspective of information overload, humans have only a limited capacity to absorb and process massive information or knowledge [6, 13]. In other words, individuals may receive enormous knowledge without effectively identifying, justifying, and understanding it. This will limit their cognitive processing ability to address and utilize the knowledge, thereby decreasing the quality of their decision-making [6, 13, 33].

Because software development is conducted in teams, if information overload occurs, it may limit and decrease the team's processing ability in terms of the recognition, analysis, application, and integration of knowledge. This is especially true when conducting knowledge-intensive SPT that may result in unilateral or fragmentary tailoring decisions. Nevertheless, with the assistance of TMS, the SPT process is facilitated by strengthening the knowledge identification capacity of expert members. Each member can concentrate on his/her domain-specific acquainted knowledge while collaboratively composing the knowledge network to produce a holistic tailoring decision. TMS is expected to lead team members in a natural way to replenish appropriate and conversant individual knowledge areas in the grand scheme of the tailoring needs. Thus, members' work can be mutually adjusted based on their complementary knowledge that is deficient in others [39]. In this sense, TMS decreases the possibility of information overload at the team level when performing SPT and enables the team to make better tailoring decisions. Thus, the following propositions are formed:

Proposition 1 TMS has a positive influence on the efficiency of SPT.

Proposition 2 TMS has a positive influence on the effectiveness of SPT.

With regard to SPT, the time remaining in a project determines how a software team reschedules and redesigns the project. In this context, STC enable team members to be in the same temporal frame regarding the length of project schedules and to have a common understanding of time-based perspectives on executing collective tasks. High levels of STC indicate that members possess equal temporal considerations [9, 22] regarding the tailoring decision, including agreement on the pacing style and meeting temporal milestones, how the work should be scheduled over time through times of process alterations, and agreement regarding when to start and finish the work on tailored tasks. Therefore, the team's "temporal dynamics" [22] are reduced, and the members can better organize and align time-related concerns to reach a decision. This improves members' shared sense of the peripheral knowledge related to the tailored tasks and subsequent tasks because members can more accurately understand others' specialized knowledge and skills in relation to these tasks at different points in time during the project. Thus, the team's TMS may be enhanced, which in turn yields a more comprehensive scheme for tailoring decisions.

Moreover, scholars have indicated that coordinative efforts are most effective when the temporal perspectives and behaviors of team members are aligned [7, 34]. In SPT, this may synchronize the timing of knowledge processing to the tailoring

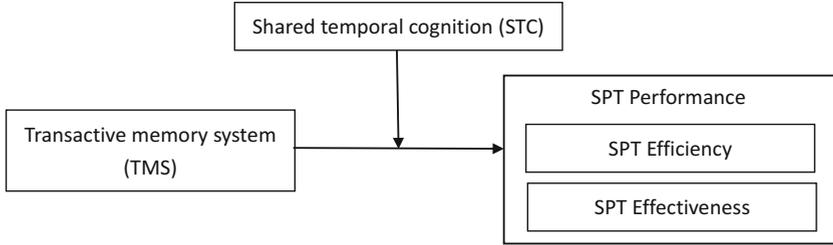


Fig. 1 The theoretical model in this paper

work. Specifically, increased coordination and time-based compatible behavioral patterns promote a flow of knowledge that composes, aggregates, diffuses, and permeates among the members, thus streamlining the discussion process during SPT. Conversely, when team members perceive time distance and time frames differently, the members may suffer mismatch and disrupted knowledge interactions, and they may have difficulty reaching a consensus regarding how tasks are adjusted and rescheduled. This may reduce the flow of knowledge transmission and transfer when performing SPT tasks. In this sense, STC improves the shared sense among members when knowledge is needed to support and contribute to the flow because members can more accurately participate in the discussion process of SPT. Thus, when a high level of shared STC exists within a software team, its TMS could be strengthened, which leads to the efficient exchange of SPT knowledge during process tailoring. In other words, the effect of the team's TMS on the efficiency of SPT could be reinforced by STC. Therefore, we propose the following:

Proposition 3 Shared temporal cognitions positively moderate the effect of a software team's TMS on SPT effectiveness.

Proposition 4 Shared temporal cognitions positively moderate the effect of a software team's TMS on SPT efficiency.

Based on the aforementioned development, a theoretical model is then established by examining the constructs of TMS, STC, and SPT performance, as shown in Fig. 1 below.

6 Concluding Remark and Future Research

This paper serves as a preliminary study of proposing a theoretical model to explore the effects of TMS and STC on SPT performance. We took the propositional approach to theorize a research model comprising several propositions basing on existing theoretical background of the three subjects. The model is conceptual in nature; thus in the next study, we would like to conduct an empirical investigation to realize the theoretical model proposed in this paper to receive a broader and

generalized aspect of the results from samples to the population of interest. Specifically, we will take the survey method, in which empirical data are collected from software development teams that have exercised and possess experiences of process tailoring. Due to the scope of the survey that is at the team level, single-source bias may occur, and thus matched pairs (i.e., different informants for different constructs) are suggested in the participant design. As for the statistical techniques for analysis, structural equation modeling (SEM) may be appropriate for testing theoretical relationships to obtain more quality results.

As for other future research, we suggest that future research extension can take the following directions. Firstly, the proposed model and propositions can be further studied using case-based approach. A case research method is useful when the empirical study expects a more in-depth exploration that is from the operational aspect. In conducting this method, we suggest that data collection should consider the issue of triangulation. That is, the empirical evidences can be collected from at least three different sources within the studied case. For example, data can be acquired from (1) archives, such as project documentations, tailoring technical reports, and customer records; (2) onsite observations on tailoring meetings, work activities, and team conversations to obtain elaborate field notes; and (3) interviews to obtain data for analysis. As for the validity of data when conducting multiple cases, it is suggested to perform a cross analysis to compare the practices across the teams to ensure the inter-consistency of the results.

References

1. L. Argote, M. Hora, Organizational learning and management of technology. *Prod. Oper. Manag.* **26**(4), 579–590 (2017)
2. A.S. Campanelli, R.D. Camilo, F.S. Parreiras, The impact of tailoring criteria on agile practices adoption: A survey with novice agile practitioners in Brazil. *J. Syst. Softw.* **137**, 366–379 (2018)
3. X. Cao, A. Ali, Enhancing team creative performance through social media and transactive memory system. *Int. J. Inf. Manag.* **39**, 69–79 (2018)
4. S.Y. Choi, H. Lee, Y. Yoo, The impact of information technology and transactive memory systems on knowledge sharing, application, and team performance: A field study. *MIS Q.* **34**(4), 855–870 (2010)
5. C. Ebert, J.D. Man, Effectively utilizing project, product and process knowledge. *Inf. Softw. Technol.* **50**(6), 579–594 (2008)
6. T. Ellwart, C. Happ, A. Gurtner, O. Rack, Managing information overload in virtual teams: Effects of a structured online team adaptation on cognition and performance. *Eur. J. Work Organ. Psy.* **24**(5), 311–317 (2015)
7. J.M.P. Gevers, M.A.G. Peeters, A pleasure working together? The effects of dissimilarity in team member conscientiousness on team temporal processes and individual satisfaction. *J. Organ. Behav.* **30**(3), 379–400 (2009)
8. J.M.P. Gevers, C.G. Rutte, W. van Eerde, Meeting deadlines in work groups: Implicit and explicit mechanisms. *Appl. Psychol. Int. Rev.* **55**, 52–72 (2006)
9. J.M.P. Gevers, W. van Eerde, C.G. Rutte, Team self-regulation and meeting deadlines in project teams: Antecedents and effects of temporal consensus. *Eur. J. Work Organ. Psy.* **18**(3), 295–321 (2009)

10. M.P. Ginsberg, L.H. Quinn, Process tailoring and the software capability maturity model. Technical Report CMU/SEI-94-TR-024. Software Engineering Institute, Pittsburgh, 1995
11. C.C. Huang, P.K. Chen, Exploring the antecedents and consequences of the transactive memory system: An empirical analysis. *J. Knowl. Manag.* **22**(1), 92–118 (2018)
12. J. Kees, Temporal framing in health advertising: The role of risk and future orientation. *J. Curr. Issues Res. Advert.* **32**(1), 33–46 (2010)
13. L.F. Laker, C.M. Froehle, J.B. Windeler, C.J. Lindsell, Quality and efficiency of the clinical decision-making process: Information overload and emphasis framing. *Prod. Oper. Manag.* (2017). <https://doi.org/10.1111/poms.12777>
14. J.C. Lee, C.Y. Chen, Investigating the environmental antecedents of organizations' intention to adopt agile software development. *J. Enterp. Inf. Manag.* **32**(5), 869–886 (2019)
15. J.C. Lee, C.Y. Chen, Exploring the team dynamic learning process in software process tailoring performance: a theoretical perspective. *J. Enterp. Inf. Manag.* (In press) (2019)
16. K. Lewis, B. Herndon, Transactive memory systems: Current issues and future research directions. *Organ. Sci.* **22**(5), 1254–1265 (2011)
17. Y.H. Li, J.W. Huang, Exploitative and exploratory learning in transactive memory systems and project performance. *Inf. Manag.* **50**(6), 304–313 (2013)
18. T.P. Liang, J. Jiang, G.S. Klein, Y.C. Liu, Software quality as influenced by informational diversity, task conflict, and learning in project teams. *IEEE Trans. Eng. Manag.* **57**(3), 477–487 (2010)
19. N. Liberman, Y. Trope, The role of feasibility and desirability considerations in near and distant future decisions: A test of temporal construal theory. *J. Pers. Soc. Psychol.* **75**(1), 5–18 (1998)
20. T. Lin, J.S. Hsu, K. Cheng, S. Wu, Understanding the role of behavioural integration in ISD teams: An extension of transactive memory systems concept. *Inf. Syst. J.* **22**(3), 211–234 (2012)
21. S. Mohammed, S. Nadkarni, Time-based individual differences and team performance: The moderating role of temporal leadership. *Acad. Manag. J.* **54**, 489–508 (2011)
22. S. Mohammed, S. Nadkarni, Are we all on the same temporal page? The moderating effects of temporal team cognition on the polychronicity diversity–team performance relationship. *J. Appl. Psychol.* **99**(3), 404–422 (2014)
23. D. Nevo, I. Benbasat, Y. Wand, Understanding technology support for organizational Transactive memory: Requirements, application, and customization. *J. Manag. Inf. Syst.* **28**(4), 69–98 (2012)
24. P.W. Paese, Effects of framing on actual time allocation decisions. *Organ. Behav. Hum. Decis. Process.* **61**(1), 67–76 (1995)
25. S.H. Park, D.H. Bae, An approach to analyzing the software process change impact using process slicing and simulation. *J. Syst. Softw.* **84**(4), 528–543 (2011)
26. S.H. Park, D.H. Bae, Tailoring a large-sized software process using process slicing and case-based reasoning technique. *IET Softw.* **7**(1), 47–55 (2013)
27. V. Peltokorpi, M. Hasu, Transactive memory systems in research team innovation: A moderated mediation analysis. *J. Eng. Technol. Manag.* **39**, 1–12 (2016)
28. Y. Ren, L. Argote, Transactive memory systems 1985–2010: An integrative framework of key dimensions, antecedents, and consequences. *Acad. Manag. Ann.* **5**(1), 189–229 (2011)
29. C.M. Santos, A.M. Passos, S. Uitdewilligen, A. Nübold, Shared temporal cognitions as substitute for temporal leadership: An analysis of their effects on temporal conflict and team performance. *Leadersh. Q.* **27**(4), 574–587 (2016)
30. S. Sawyer, Effects of intra-group conflict on packaged software development team performance. *Inf. Syst. J.* **11**(2), 155–178 (2001)
31. B. Simeonova, Transactive memory systems and web 2.0 in knowledge sharing: A conceptual model based on activity theory and critical realism. *Inf. Syst. J.* **28**(4), 592–611 (2017)
32. L. Slaughter, P.-H. Ramesh, Baskerville, Aligning software processes with strategy. *MIS Q.* **30**(4), 891–918 (2006)
33. C. Speier, J.S. Valacich, I. Vessey, The influence of task interruption on individual decision making: An information overload perspective. *Decis. Sci.* **30**(2), 337–360 (1999)

34. R.L. Standifer, A.M.L. Raes, C. Peus, A.M. Passos, C.M. Santos, S. Weisweiler, Time in teams: Cognitions, conflict and team satisfaction. *J. Manag. Psychol.* **30**(6), 692–708 (2015)
35. J.F. Tripp, D.J. Armstrong, Agile methodologies: Organizational adoption motives, tailoring, and performance. *J. Comput. Inf. Syst.* **58**(2), 170–179 (2018)
36. Y. Trope, N. Liberman, Temporal construal. *Psychol. Rev.* **110**(3), 403–421 (2003)
37. S. Uitdewilligen, M.J. Waller, A.H. Pitariu, Mental model updating and team adaptation. *Small Group Res.* **44**, 127–158 (2013)
38. Y. Wang, Q. Huang, R.M. Davison, F. Yang, Effect of transactive memory systems on team performance mediated by knowledge transfer. *Int. J. Inf. Manag.* **41**, 65–79 (2018)
39. E. Whelan, R. Teigland, Transactive memory systems as a collective filter for mitigating information overload in digitally enabled organizational groups. *Inf. Organ.* **23**(3), 177–197 (2013)
40. P. Xu, B. Ramesh, Software process tailoring: An empirical investigation. *J. Manag. Inf. Syst.* **24**(2), 293–328 (2007)
41. P. Xu, B. Ramesh, Impact of knowledge support on the performance of software process tailoring. *J. Manag. Inf. Syst.* **25**(3), 277–314 (2008)

Mixed-Integer Linear Programming Model for the Simultaneous Unloading and Loading Processes in a Maritime Port



Ali Skaf, Sid Lamrous, Zakaria Hammoudan, and Marie-Ange Manier

1 Introduction and Literature Review

The general process in a container terminal can be described as a sequence of operations from the arrival to the departure of the container's vessels. The container vessel is dedicated to transport containers from a maritime port to another. The container is a parallelepiped metal box designed for the transport of goods by different modes of transport. The quay crane is used to load containers into or unload containers from the vessel. Yard trucks are used for transporting containers from the station of quay cranes to the storage location or vice versa. In the storage location, there is a type of crane called reach-stacked crane, it can be used to load containers into or unload containers from the yard trucks. In this study, we have two container vessels, the first one is dedicated for the containers to be unloaded and transported to the storage location (U-containers), and the second one is dedicated to the containers to be unloaded from the storage location and loaded into the vessel (L-containers). There are two separated storage locations, one for the U-containers and one for the L-containers. More precisely, the unloading process includes the following steps:

1. A quay crane unloads the U-container from the vessel and loads it into a yard truck.
2. A yard truck transports the U-container to the storage location.
3. A reach-stacker crane unloads the U-container from the yard truck and loads it in the storage location.

A. Skaf (✉) · S. Lamrous · M.-A. Manier
Univ. Bourgogne Franche-Comté, FEMTO-ST Institute/CNRS (UTBM), Belfort, France
e-mail: ali.skaf@utbm.fr; sid.lamrous@utbm.fr; marie-ange.manier@utbm.fr

Z. Hammoudan
Jinan University (JUT), Tripoli, Lebanon

After this process, the yard truck continues its way to the storage location dedicated for the L-containers. The loading process includes the following steps:

1. A reach-stacker crane collects a L-container from storage location and loads it into the yard truck coming from the storage location for the U-containers.
2. A yard truck transports the L-container to the quay crane station.
3. A quay crane unloads the L-container from the yard truck and loads it into the vessel.

Figures 1 and 2 describe the full unloading and loading operations.

In our previous studies, Skaf et al. [1] proposed a mixed-integer linear programming model and a dynamic programming algorithm to solve the quay crane scheduling problem at port of Tripoli-Lebanon. Later, Skaf et al. [2] proposed a new genetic algorithm to solve the problem, due to the inability to provide results from the two previous exact methods.

After that, Skaf et al. [3] proposed a mixed-integer linear programming model and a dynamic programming algorithm to solve the scheduling problem for single quay crane and multiple yard trucks at port of Tripoli-Lebanon.

This study is considered new to the literature, but we addressed some researchers who solved the scheduling problem for the quay cranes, the yard trucks or for both of them.

Daganzo [4] studied the quay crane scheduling problem for multiple vessels. He considered that each vessel is divided into many bays, and each bay contains a number of containers. His objective is to reduce the cost of delay using an approximate and an exact method. Furthermore, Peterkofsky and Daganzo [5] proposed a branch and bound method for the quay crane scheduling problem in the case of quay cranes crossing. After that, Kim and Park [6] explored the quay

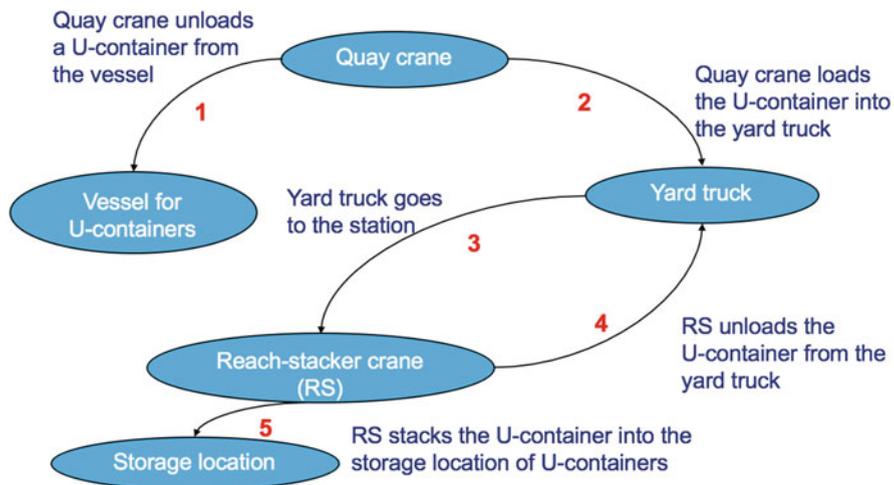


Fig. 1 U-container unloading process

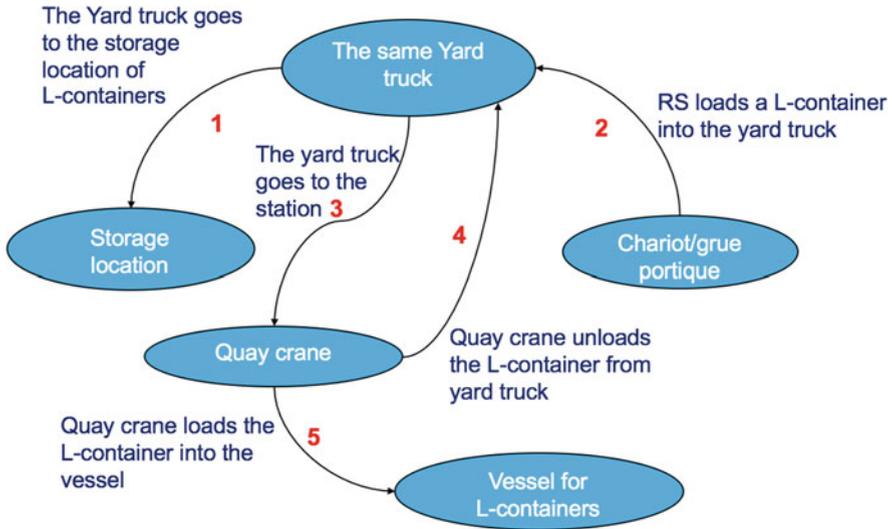


Fig. 2 L-container loading process

crane scheduling problem with non-crossing constraints, and they considered that only one quay crane can work into the vessel. Their objective was to minimize the total completion time.

Lim et al. [7] considered that each vessel is a job and each quay crane is assigned to this job. They developed a dynamic programming algorithm with a taboo search method to solve the problem. Steeken and Stahlbock [8] also studied the quay crane scheduling problem and they classified and described the logistic processes and present a new survey for their optimization. Homayouni et al. [9] proposed a genetic algorithm to schedule the quay cranes with integration of automated guided vehicles (AGV). Moreover, Diabat and Theodorou [10] proposed a formulation for the scheduling problem and all assignments for the quay cranes such as quay crane’s position. They developed a genetic algorithm to solve this problem. Furthermore, Kaveshgar et al. [11] proposed a mixed integer programming model for quay cranes and yard trucks scheduling. They also developed a genetic algorithm with a greedy search method. After that, Al-Dhaheri and Diabat [12] defined the sequence for the unloading operations by fixing a number of quay cranes to perform it. They proposed a mixed-integer programming (MIP) formulation for this problem. Finally, Vahdani et al. [13] aimed to combine the quay cranes and yard truck assignments among them. For this problem, they proposed a bi-objective optimization model.

This study proposes a mixed-integer programming model solved by CPLEX for jointly quay crane and yard truck scheduling problem where both loading and unloading operations are considered. After that, we generated results and tested our model for small and large instances, and a comparison with real results from the port of Tripoli-Lebanon.

In Sect. 2 we propose a mixed-integer linear programming model. In Sect. 3, we provide the results of the proposed model. Finally, in Sect. 4, we give a conclusion and a step for future works.

2 Mathematical Formulation

2.1 Assumptions

- The required times for loading and unloading the containers by quay cranes and reach-stacker cranes are known, so as the required times to transport containers and the positions of the containers in the vessels.
- Each quay crane can operate in a single container ship at a time.
- Each vessel can be handled by one or more quay cranes at a time.
- The priority of the containers is taken into account and there maybe a time it waits for quay cranes and yard trucks (they both expect one another).
- Each truck can transport only one container at a time.
- Each reach-stacker crane can unload/load only one container at a time.
- Each container can be transported by only one yard truck at a time.
- Each container can be unloaded/loaded by only one quay crane at a time.
- All containers are homogeneous (same size).
- We do not consider the number of reach-stacker cranes.

2.2 Notations

- Q Set of quay cranes that will unload containers from the vessel.
- Q' Set of quay cranes that will load containers to the vessel.
- T Set of yard trucks.
- C Set of containers to be unloaded from the vessel, c is the number of containers.
- C' Set of containers to be loaded to the vessel, c' is the number of containers.
- p_i Position of container i in the vessel 1, $\forall i \in C$.
- p'_i Position of container i in the vessel 2, $\forall i \in C'$.
- v Yard truck time from vessel 2 to vessel 1.
- v'_i Yard truck time from vessel 1 to the yard location for unloaded containers which exists container i , $\forall i \in C$.
- v''_i Yard truck time from yard location for unloaded containers which exists container i to the vessel 1, not loaded by any container $\forall i \in C$.
- v'''_i Yard truck time from yard location for containers to be loaded which exists container i to the vessel 2, $\forall i \in C'$.

- λ_{ij} Yard truck time from yard location for unloaded containers which exists container i to yard location for containers to be loaded which exists container j , $\forall i \in C$ and $\forall j \in C'$.
- d_i Quay crane unloading time of container i , $\forall i \in C$.
- d'_i Quay crane loading time of container i , $\forall i \in C'$.
- rs Unloading time of a container by RS.
- rs' Loading time of a container by RS.
- s_0 Distance between quay cranes for safety reason.
- $\Omega 1$ Set of precedence containers to be unloaded.
- $\Omega 2$ Set of precedence containers to be loaded.
- u One unit moving time for the quay crane.
- M Big integer.

2.3 Decision Variables

Boolean Variables

- $X_{ijq} = \begin{cases} 1 & \text{if quay crane } q \text{ unloads U-container } i \text{ before U-container } j \\ 0 & \text{otherwise, } \forall i \in \{0, \dots, c\}, \forall j \in \{1, \dots, c + 1\}, \forall q \in Q \end{cases}$
- $Y_{ijq} = \begin{cases} 1 & \text{if quay crane } q \text{ loads L-container } i \text{ before U-container } j \\ 0 & \text{otherwise, } \forall i \in \{0, \dots, c'\}, \forall j \in \{1, \dots, c' + 1\}, \forall q \in Q' \end{cases}$
- $H_{ii'} = \begin{cases} 1 & \text{if U-container } i \text{ is matched with L-container } i' \\ 0 & \text{otherwise, } \forall i \in C, \forall i' \in C' \end{cases}$
- $Z_{ijt} = \begin{cases} 1 & \text{if yard truck } t \text{ transport U-container } i \text{ before U-container } j \\ 0 & \text{otherwise, } \forall i \in \{0, \dots, c\}, \forall j \in \{1, \dots, c + 1\}, \forall t \in T \end{cases}$
- $W_{ij} = \begin{cases} 1 & \text{if round operation time of U-container } i \text{ finishes before the starts of} \\ & \text{round operation of U-container } j \text{ by the quay crane} \\ 0 & \text{otherwise, } \forall i \in C, \forall j \in C \end{cases}$
- $W'_{ij} = \begin{cases} 1 & \text{if round operation time of L-container } i \text{ finishes before the starts of} \\ & \text{round operation of L-container } j \text{ by the quay crane} \\ 0 & \text{otherwise, } \forall i \in C', \forall j \in C' \end{cases}$

Float Variables

- E_i The time when the process of U-container i ends, $\forall i \in C$
- E'_i The time when the process of L-container i ends, $\forall i \in C'$
- HA'_i Time when U-container i is ready to be transported by the yard truck, $\forall i \in C$
- HA''_i Time when L-container i is ready to be transported by the yard truck, $\forall i \in C'$

- HA_i''' Time when L-container, which is matched with U-container i , is ready to be transported by the yard truck, $\forall i \in C$
- C_{max} Makespan for both vessels loading and unloading

2.4 Modeling

The following is a mixed-integer linear programming model that we propose for the quay crane and yard truck scheduling:

Objective

$$\text{minimize } C_{max} \quad (1)$$

Equation (1) is the objective function which aims to minimize the completion time.

Subject to

$$\sum_{j=1}^{c+1} X_{0jq} = 1 \quad \forall q \in Q \quad (2)$$

$$\sum_{i=0}^c X_{i(c+1)q} = 1 \quad \forall q \in Q \quad (3)$$

$$\sum_{q \in Q} \sum_{j=1}^{c+1} X_{ijq} = 1 \quad \forall i \in C \quad (4)$$

$$\sum_{j=1}^{c+1} X_{ijq} = \sum_{j=0}^c X_{jiq} \quad \forall i \in C, \forall q \in Q \quad (5)$$

Constraints (2), (3), (4) and (5) define the sequence of unloading for U-containers by quay cranes (which ensures that the 1st U-container must be unloaded from the vessel as well as the last U-container, and ensures that all quay crane-container assignments for unloading are made).

$$\sum_{j=1}^{c'+1} Y_{0jq} = 1 \quad \forall q \in Q' \quad (6)$$

$$\sum_{i=0}^{c'} Y_{i(c'+1)q} = 1 \quad \forall q \in Q' \tag{7}$$

$$\sum_{q \in Q'} \sum_{j=1}^{c'+1} Y_{ijq} = 1 \quad \forall i \in C' \tag{8}$$

$$\sum_{j=1}^{c'+1} Y_{ijq} = \sum_{j=0}^{c'} Y_{j iq} \quad \forall i \in C', \forall q \in Q' \tag{9}$$

Constraints (6), (7), (8) and (9) define the sequence of loading for L-containers by quay cranes (which ensures that the 1st L-container must be loaded in the vessel as well as the last L-container, and ensures that all crane-container assignments for loading are made).

$$\sum_{i=1}^{c+1} Z_{0it} = 1 \quad \forall t \in T \tag{10}$$

$$\sum_{i=0}^c Z_{i(c+1)t} = 1 \quad \forall t \in T \tag{11}$$

$$\sum_{t \in T} \sum_{j=1}^{c+1} Z_{ijt} = 1 \quad \forall i \in C \tag{12}$$

$$\sum_{j=1}^{c+1} Z_{ijt} = \sum_{j=0}^c Z_{jit} \quad \forall i \in C, \forall t \in T \tag{13}$$

Constraints (10), (11), (12) and (13) provide the transport sequence of the U-containers from the vessel by the yard trucks.

$$\sum_{i' \in C'} H_{ii'} = 1 \quad \forall i \in C \tag{14}$$

$$\sum_{i \in C} H_{ii'} = 1 \quad \forall i' \in C' \tag{15}$$

Constraints (14), (15) give all the unique assignment for the pairs of U-containers and L-containers which correspond to each other.

$$E_i \geq d_i - M * (1 - \sum_{q \in Q} X_{0iq}) \quad \forall i \in C \quad (16)$$

$$E_i \geq E_j - M * (1 - \sum_{q \in Q} X_{j iq}) + (p_i - p_j) * u + d_i \quad \forall i \in C, \forall j \in C \quad (17)$$

Constraints (16), (17) provide the completion time for unloading the U-containers by the quay cranes from the vessel.

$$E'_i \geq E'_j - M * (1 - \sum_{q \in Q'} X_{j i q}) + (p'_i - p'_j) * u + d'_i \quad \forall i \in C', \forall j \in C' \quad (18)$$

$$E'_i \geq HA''_i + d'_i \quad \forall i \in C' \quad (19)$$

Constraints (18), (19) provide the completion time for loading L-containers by quay cranes into the vessel.

$$HA'_i \geq E_i + v'_i + rs \quad \forall i \in C \quad (20)$$

$$HA'_i \geq HA'''_j - M * (1 - \sum_{t \in T} Z_{j it}) + v'_i + rs \quad \forall i \in C, \forall j \in C \quad (21)$$

Constraints (20), (21) provide the completion time for transporting the U-containers by the yard trucks.

$$HA''_{i'} \geq HA'_i - M * (1 - H_{ii'}) + \lambda_{ii'} + rs' + v''_{i'} + v \quad \forall i \in C, \forall i' \in C' \quad (22)$$

Constraint (22) provides the completion time for transporting L-containers by yard trucks.

$$HA'''_i \geq HA''_{i'} - M * (1 - H_{ii'}) \quad \forall i \in C, \forall i' \in C' \quad (23)$$

Constraint (23) provides the completion time to transport the L-container which is matched with the U-container, by the yard truck.

$$E_j - d_j \geq E_i \quad \forall (i, j) \in \Omega 1 \quad (24)$$

Constraint (24) ensures that the operation of each U-container must be completed before another U-container that follows it, if they belong to $\Omega 1$.

$$E'_j - d'_j \geq E'_i \quad \forall (i, j) \in \Omega 2 \quad (25)$$

Constraint (25) ensures that the operation of each L-container must be completed before another L-container that follows it, if they belong to Ω_2 .

$$M * (1 - W_{ij}) \geq E_i - (E_j - d_j) \quad \forall i \in C, \forall j \in C \quad (26)$$

$$M * (1 - W'_{ij}) \geq E'_i - (E'_j - d'_j) \quad \forall i \in C', \forall j \in C' \quad (27)$$

$$E_j - d_j - E_i \leq M * W_{ij} \quad \forall i \in C, \forall j \in C \quad (28)$$

$$E'_j - d'_j - E'_i \leq M * W'_{ij} \quad \forall i \in C', \forall j \in C' \quad (29)$$

$$(p_i - p_j) * (i - j) + M * (W_{ij} + W_{ji}) \geq (i - j) * s_0 \quad \forall i \in C, \forall j \in C, i \neq j \quad (30)$$

$$(p'_i - p'_j) * (i - j) + M * (W'_{ij} + W'_{ji}) \geq (i - j) * s_0 \quad \forall i \in C', \forall j \in C', i \neq j \quad (31)$$

Constraints (26), (27), (28), (29), (30) and (31) guarantee the non-crossing and the safety margin between the quay cranes.

$$C_{max} \geq E'_i \quad \forall i \in C' \quad (32)$$

Constraint (32) indicates the completion time of the vessel who contains the U-containers.

In the previous model, we suppose that the number of U-containers is equal to the number of L-containers. Nevertheless, the numbers of U-containers and L-containers are different. For the case where the number of U-containers is bigger than the number of L-containers, there are no L-containers that matched with the U-containers and the yard truck will return empty from the storage location. So for this reason we will add $c - c'$ fictive L-containers.

For this case, we propose a model extension and it is formulated as follows:

Objective

$$\text{minimize } C_{max} \quad (33)$$

Subject to

constraints (2) \rightarrow (14), (16) \rightarrow (21), (23) \rightarrow (31)

$$\sum_{i \in C} H_{ii'} = 1 \quad \forall i' \in \{1, \dots, c', \dots, c\} \quad (34)$$

Constraint (34) provides all the unique assignments for the pairs including the fictive L-containers.

$$HA''_{i'} \geq HA'_i - M * (1 - H_{ii'}) + v''_i$$

$$\forall i \in C \quad \forall i' \in \{c' + 1, \dots, c'\} \quad (35)$$

Constraint (35) presents the completion time of the U-containers with the empty movements.

$$C_{max} \geq HA_i''' \quad \forall i \in C \quad (36)$$

Constraint (36) defines the makespan of all arriving vessels.

In another way, we swapped the constraint (15) by (34), the constraint (22) by (35) and the constraint (32) by (36).

3 Experimental Results

The model is solved using the CPLEX 12.6 solver, and the tests are run on MacBook Pro 2.7 GHz Intel Core i5 with 8GB RAM 1867 MHz DDR3 under OSX 10.11.6.

In this section, we are presenting the results generation and the results obtained for real cases in the port of Tripoli-Lebanon. The makespan is measured in time units (u.t).

3.1 Results for Randomly Generated Instances

Table 1 shows the results of the calculation tests when the numbers of U-containers and L-containers are the same, and when the number of U-containers is greater than the L-containers. For example in instance 24 in Table 1, for 30 U-containers, 25 L-containers, 7 quay cranes (jointly for unloading and loading) and 10 yard trucks, CPLEX cannot provide any result after 3 hours of execution, then we interrupt the execution (N.A. = Interrupt execution (No results)). In this table, we notice that the proposed MILP works for small and medium instances and does not work so well for large instances. So in our next work, we will propose a new exact or metaheuristic methods to improve the execution time and obtain near optimal solutions.

3.2 Results for Real Instances from Port of Tripoli-Lebanon

Table 2 compares some results from the port of Tripoli-Lebanon, with the obtained results by this model . We emphasize that all port's results are considered in the same values and conditions of the port of Tripoli-Lebanon. As shown in Table 2, our model succeeded in improving the completion time of containers, for all the tested instances, by an average 20%. $GAP(\%) = ((\text{port result} - \text{CPLEX result})/\text{port result}) * 100$.

Table 1 Experimental results

Instance	C	C'	Q	Q'	T	CPLEX	Instance	C	C'	Q	Q'	T	CPLEX
1	6	6	1	1	1	1115	14	15	15	2	2	4	562
2	6	4	1	1	2	492	15	16	12	3	2	6	312
3	8	8	1	1	2	748	16	16	14	3	2	6	433
4	8	6	2	2	3	433	17	16	14	3	2	7	327
5	10	10	2	2	4	467	18	18	14	4	3	8	340
6	10	6	2	2	3	527	19	18	16	4	3	10	337
7	10	8	2	2	4	450	20	18	16	4	3	8	229
8	12	8	2	2	4	445	21	20	20	2	3	6	476
9	12	10	2	2	4	545	22	25	20	4	3	10	229
10	12	12	2	2	5	452	23	26	24	4	3	10	225
11	12	10	2	2	5	351	24	30	25	4	3	10	N.A
12	14	12	2	2	5	382	25	30	30	2	2	6	N.A
13	14	12	3	2	6	270							

Table 2 Comparison with the real results in port of Tripoli-Lebanon

Instance	Q	Q'	C	C'	T	Port results (s)	CPLEX results (s)	GAP (%)
P1	1	1	2	2	1	783	629	19.67
P2	1	1	5	4	2	965	782	18.96
P3	1	1	6	6	2	1129	912	19.22
P4	1	1	7	4	2	1046	835	20.17
P5	1	1	8	7	2	1393	1076	22.76

4 Conclusion

This model investigates the scheduling problem for the quay cranes with yard trucks in an integrated way. We use the dual strategies to reduce the empty movements for the yard trucks. We proposed a mixed-integer linear programming model to minimize the completion time of all containers in the vessels, and thus reducing the docking time of all vessels. From the numerical results, we can see that the proposed model is feasible. For small instances, CPLEX provides results with an acceptable execution time. But for larger instances, CPLEX cannot provide any result. So, in our future studies, we will develop exact or metaheuristic algorithms to compare operational results and thus obtain results for large instances.

References

1. A. Skaf, S. Lamrous, Z. Hammoudan, M.-A. Manier, Exact method for single vessel and multiple quay cranes to solve scheduling problem at port of tripoli-lebanon, in *International Conference on Industrial Engineering and Engineering Management (IEEM)* (2018)
2. A. Skaf, S. Lamrous, Z. Hammoudan, M.-A. Manier, Genetic algorithm to optimize unloading of large containers vessel in port of tripoli-lebanon, in *International Conference on Control, Decision and Information Technologies (CODIT)* (2019)
3. A. Skaf, S. Lamrous, Z. Hammoudan, M.-A. Manier, Single quay crane and multiple yard trucks scheduling problem with integration of reach-stacker cranes at port of tripoli-lebanon, in *International Conference on Systems, Man, and Cybernetics (SMC)* (2019)
4. C. Daganzo, The quay crane scheduling problem. *Transportation Research* **23**, 159–175 (1989)
5. R. Peterkofsky, C. Daganzo, A branch and bound solution method for the quay crane scheduling problem. *Transportation Research* **24**, 159–172 (1990)
6. K. Kim, Y. Park, A crane scheduling method for port container terminals. *Eur. J. Oper. Res.* **156**, 752–768 (2004)
7. F. Xiao, A. Lim, B. Rodrigues, Y. Zhu, Quay crane scheduling with spatial constraints. *Naval Res. Logist.* **51**, 386–406 (2004)
8. D. Steeken, R. Stahlbock, Container terminal operation and operations research - classification and literature review. *OR Spectrum* **26**, 3–49 (2004)
9. S.M. Homayouni, S.H. Tang, O. Motlagh, A genetic algorithm for optimization of integrated scheduling of cranes, vehicles, and storage platforms at automated container terminals. *J. Comput. Appl. Math.* **270**, 545–556 (2013)
10. A. Diabat, E. Theodorou, An integrated quay crane assignment and scheduling problem. *Comput. Ind. Eng.* **73**, 115–123 (2014)
11. N. Kaveshgar, N. Huynh, Integrated quay crane and yard truck scheduling for unloading inbound containers. *Int. Prod. Econ.* **159**, 168–177 (2014)
12. N. Al-Dhaheer, A. Diabat, The quay crane scheduling problem. *J. Manuf. Syst.* **36**, 87–94 (2015)
13. B. Vahdani, F. Mansour, M. Soltani, D. Veysmoradi, Bi-objective optimization for integrating quay crane and internal truck assignment with challenges of trucks sharing. *Knowl. Based Syst.* **163**, 675–692 (2018)

How to Test Interoperability of Different Implementations of a Complex Military Standard



Andre Schöbel , Philipp Klotz , Christian Zschke ,
and Barbara Essendorfer 

1 Introduction

In nowadays military defense coalitions it is inevitable to standardize the used material and equipment as well as the applied procedures among the partners and units. This enables an efficient cooperation in a coalition and between collaborating troops. To establish these guidelines and procedures in the NATO (North Atlantic Treaty Organization) environment, Standardization Agreements (STANAGs) were defined and ratified. These documents were developed and continuously adapted taking into account changes in technology as well as in the requirements specified by international working groups called Custodian Support Teams (CSTs). Those teams consist of experts from nations and organizations that have an interest in a specific STANAG. The connection between many STANAGs used in the context of Joint Intelligence, Surveillance, and Reconnaissance (JISR) and how they are working together is specified in the NATO Intelligence, Surveillance, and Reconnaissance Interoperability Architecture (NIIA) [12].

Although the standards were created with the intention to define a clear specification and guideline, some parts of these documents are described very superficially and with several ambiguities. This is due to finding a common agreement between the different nations and due to the general issue that specifying in natural language is often ambiguous. Especially in the usage of standards that result in a software system, these ambiguities can lead to a wide range of possible implementations among the different software providers. Nevertheless, the interoperability between

A. Schöbel (✉) · P. Klotz · C. Zschke · B. Essendorfer
Fraunhofer IOSB, Karlsruhe, Germany
e-mail: andre.schoebel@iosb.fraunhofer.de; philipp.klotz@iosb.fraunhofer.de;
christian.zschke@iosb.fraunhofer.de; barbara.essendorfer@iosb.fraunhofer.de
<http://www.iosb.fraunhofer.de>

the resulting heterogeneous systems must be established. To meet this requirement, it is necessary to test the systems and their interactions with each other before going into common usage.

This can be done in two ways:

Software Testing With software tests, the software providers validate the desired behavior and the correctness of their developed software. According to test-driven development (TDD) [4], during the implementation of a software, test cases need to be defined and implemented by the developers. Then, the tests can be used to validate that newly created functionalities are working correctly and do not break the already existing code.

Therefore, the tests are usually executed periodically and fully automatically several times a day using continuous integration and continuous delivery (CI/CD) [14] tools and mechanisms. The creation and the continuous maintenance of the tests can only be achieved by investing a lot of time and money. Nevertheless, it cannot be guaranteed that all characteristics and cases of failures are covered. Additionally, those tests are also depending on the interpretation of the specification. This results in the fact that a system can be successfully tested internally but nevertheless be unable to exchange data with other systems when going into operation.

Military Exercises and Trials In the military environment, it is also common to arrange exercises and trials to simulate various scenarios. These events are used to train and validate predefined operational scenarios and workflows as well as the interoperability between the systems in use. The “Coalition Warrior Interoperability eXercise (CWIX)” [1] and the trial “Unified Vision” are two events in the NATO environment where different operators and software providers come together to collaborate and verify the ability to work in cooperation.

Especially the trials have an operational focus meaning that, as a prerequisite, systems need to cooperate and interoperability must be ensured beforehand. High costs and a long-time interval between the exercises are the disadvantages of this kind of possibility to validate the correctness of a system. To derive the most benefit of such an exercise, it is essential that all participating systems are working correctly and can be used to fulfill the scenario. In the past, that has not always been the case. Time was wasted to find general bugs and malfunctions especially with regard to the interoperability between the different implementations.

These problems led to considerations to establish a central and independent testing facility providing standardized test cases that can be used by the software providers and developers to validate the conformance of their implementations to the STANAG 4559 standard [11].

Within this chapter, an overview of the underlying documents that can be considered as a source for test cases is provided. Then, the analysis of these documents toward the definition of concrete test cases and requirements is described followed by a section that lists some general aspects and ideas regarding a testing facility. The chapter closes with an overview and possible next steps.

2 Fundamentals

One of the central elements of the NIIA is the STANAG 4559. This standard defines interfaces, data models, and workflows to disseminate artifacts within the JISR process. Since the focus of a testing facility should be on testing the compliance of implementations to the STANAG 4559, the standard itself and its related documents, especially the AEDP-17 [8], form the foundation for the test cases. To provide the needed background for the main chapter, a short introduction into the observed documents will be given in the following sections.

2.1 *JISR Process*

In multinational operations that include all forces it is necessary to define interoperable processes to be able to exchange information efficiently. The JISR process aims at synchronizing and integrating the planning and operation of data collection capabilities from different intelligence disciplines with the processing, exploitation, and dissemination of the resulting information. In AJP-2.7 [6] and AIntP-14 [7] this process is described, and relevant roles and activities are specified. To be able to support this process with technical means it is important to also define standards, like the STANAG 4559, and workflows with a technical perspective.

2.2 *STANAG 4559*

To fulfill a military mission target, different systems are combined in a so-called System of Systems (SOS) architecture. Thereby, each system provides its features to the coalition, so that in combination of all systems an advantage for all participants can be created. In the military surveillance and reconnaissance environment, this concept is already applied by connecting different planning, tasking, sensor, and exploitation systems together with data storage and dissemination components to a large SOS. In the NATO environment, this is called “Coalition Shared Data (CSD) concept.” Different information artifacts can be exchanged through common services, interfaces, and data models. The concept includes the dissemination of artifacts that are relevant for planning operations within the Information Requirements Management and Collection Management (IRM&CM) [13] process and are relevant for initializing the JISR process. These artifacts are shared using services described in NATO - AEDP-19 - NATO Standard ISR Workflow Architecture [10].

Products like images, videoclips, and reports are stored in combination with additional metadata information like geographic references, creation date, and publisher. It is also possible to link the products to each other via associations, e.g., the relation between an image and a derived report can be defined. One part

of this concept is called CSD-Server and is described in NATO - AEDP-17 - NATO Standard ISR Library Interface [8]. The CSD-Server provides a standardized interface to store standardized products (referenced in the JISR process as JISR results) with their corresponding metadata and to search for them via query or subscription.

To disseminate the data in the coalition network, a synchronization functionality is also defined in the standard. Thereby, the CSD metadata is shared across multiple CSD-Servers over a Wide Area Network (WAN) in consideration of different releasability and security requirements. In order to reduce the bandwidth consumption, the actual products are only transferred when requested. To search and use the stored data, a client system can be connected to a CSD-Server in the network. A CSD-Client allows to retrieve and update the stored products or to ingest completely new products.

In the chapters “Evolution of the Coalition Shared Data concept in Joint ISR” [2] and “Adaptation of interoperability standards for cross domain usage” [3] the whole concept is described in more detail and several restrictions of the standard are outlined.

Similar to the concept of the CSD-Server, the CSD-Streaming Server enables the sharing of data streams as videos or tracks. The streaming capability is described in NATO - AEDP-18 - NATO Standard ISR Streaming Services [9]. All mentioned AEDPs (AEDP-17, AEDP-18, AEDP-19) are part of the STANAG 4559 in its current version [11]. As a starting point to validate and test the conformity of an implementation, the AEDP-17 describes the “Abstract Conformance Test Suite” (A.C.T.S.) which defines a collection of abstract conformance test cases [8, see ANNEX L]. Each standard-compliant implementation must be able to pass these test cases. The test cases were defined considering the primary and secondary actors described in the AEDP-17 [8, see Chapter B-2.2] and can be used to test the basic functions and requirements defined by the standard.

Basically the test cases are separated into two main groups:

Conformance Testing for CORBA Interface

These tests are defined to validate the Common Object Request Broker Architecture (CORBA) interface implementation.

Considered test cases according to [8, Chapter L-2]:

- “Ingest Catalogue Entry”
- “Search and Subscription”
- “Retrieval of Files”
- “Updating Metadata”
- “Retrieval of Data Model”
- “Mark Metadata Obsolete”
- “Synchronize with Another Server”

Conformance Testing for Web Service Interfaces

These tests are defined to validate the implementation of the web service interfaces. Considered test cases according to [8, Chapter L-3]):

- “Ingest Catalogue Entry”
- “Updating Metadata”
- “Mark Metadata Obsolete”
- “Search and Incremental queries”

3 Test Center

The STANAG 4559 documentation is the result of a long-time, evolutionary process. It defines the requirements for different subjects of the exchange and provision process of data between NATO nations and their partners on different levels of abstraction and with a greater or lesser extent of detail. The varying quality of the specification is the main challenge in testing systems that implement this standard. The inherent ambiguity in the specification makes it almost impossible to test the standard in detail without establishing a broad agreement in every single detail between all stakeholders involved. This agreement has not been achieved yet and also is hard to be achieved in the future.

Taking into account the liabilities of the standard, it is necessary to achieve a certain depth of tests without interpreting ambiguous finer points of the specification. A certain test depth is crucial for the relevance and validity of the test results. The absence of subjective interpretation, e.g., of one provider, is essential for the acceptance of the tests in the STANAG 4559 community.

To provide an opportunity for all interested stakeholders to test their implementations, the idea of creating an independent testing facility emerged. The testing facility “JISR Test Center” aims to provide meaningful tests to prove standard conformity of specific implementations provided by customers, i.e., software providers.

These tests could serve as a precondition for the participation in exercises and trials or to validate the standard conformance before putting a system into operation. The tests could also help to identify potential weaknesses in the implementation and in the tests, e.g., concerning test coverage, of a customer’s system. As another aspect, the tests potentially uncover deficiencies in the specification of the standard itself. It could therefore help to start a constructive debate about weaknesses of the standard and to enhance the overall quality of the specification and derived tests.

The JISR Test Center purposes to allow easy test attendance and to provide meaningful and comprehensible test results. To achieve easy participation in testing, the tests should be executable at the JISR Test Center, i.e., on-site, as well as remotely. Test results should be written in a comprehensible language and with regard to a clear presentation.

Because the JISR Test Center, as a neutral authority, will be operated by testers that have not necessarily studied or implemented the standard in all details, the JISR Test Center should be easy to operate. The tests should also be well documented. To reduce the complexity of the standard, the description of specific aspects of the standard should be abstracted and simplified wherever possible and useful. A high degree of automation of the JISR Test Center tests helps reducing the execution time and the effort of conducting complex tests.

The following section describes the demands, challenges, and solution approaches for the realization of tests for the JISR Test Center.

4 Testing

4.1 Requirements on the Tests

As mentioned above, the JISR Test Center is intended to be a central and independent testing facility for the stakeholders of the different implementations of the STANAG 4559 systems. Tests must be executed and understood by testers without deep understanding of programming or fine granular knowledge of the STANAG 4559. Therefore, the requirements on testing, described in the following, mainly concern the usability of tests.

Before starting the tests, the essential information, needed by the tests to be executed, has to be provided by the testers. The tests then should run fully automatically without involving the testers until all tests have been executed.

The testers should be able to easily follow the progress of the tests and view the previous results of all tests in one perspective. Testers should have easy access on details of test results for later analysis. Details of a test should reflect information about what was done in a test. The detailed information should ideally provide the reason why a test failed. When all tests have been executed, a summary of all tests should be given. The test results and the summary should be supplied in one place so that testers are able to easily access them as a whole. In the best case, testers can do other work during automatic test procedures.

4.2 How to Get Test Cases

The first considered source for tests was one of the annexes of the STANAG 4559 and the therein specified Abstract Conformance Test Suite (A.C.T.S.) [8, see ANNEX L]. The scope of tests defined in this annex are testing the use cases of systems which are implementing the AEDP-17 [8, see 2.2 USE CASES]. Use cases are, for example, the ingest of products or associating products with each other.

The A.C.T.S.-tests are defined on a technical level, that means that they specify the CORBA—and web service interface methods that should be called. The general intention of these tests is that the interface method calls can be made, but neither that they are answered correctly nor that they have been correctly executed.

Consider the following specification of a test for the CORBA interface which scope is to determine the amount of products, respectively, the amount of associations in the system to be tested [8, see L-2.2.2 Searching for hit count]:

L-2.2.2 Test Case “Searching for hit count”

Perform the following sequence of steps for both products and associations:

1. get Library instance
2. get CatalogMgr instance
3. invoke CatalogMgr::hit_count
4. invoke HitCountRequest::complete
5. clean up request
 - (a) invoke HitCountRequest::cancel
 - (b) invoke CatalogMgr::delete_request

The test specification states the CORBA interface methods that must be invoked in sequence to set up and clean up the actual test method—the hit_count method. It does not make any constraint on the outcome of the hit_count method. The integer returned is not supposed to be compared to the real amount of products, respectively, associations in the system to be tested, which is from now on called “customer system.” The method may return any integer.

As another example, consider the following specification of a test for the web service interface where the scope is to update the metadata of products and associations [8, see L-3.2 Test Case “Updating Metadata”]:

L-3.2 Test Case ‘Updating Metadata’

Perform the following sequence of steps for both products and associations:

1. Get CSD-Publish interface
2. Invoke update on the CSD-Publish interface with the new metadata content

The test specification states that an update method should be invoked on the CSD-Publish interface with new metadata content for products and associations. The test neither specifies the exact attributes nor their values for the metadata content that should be updated. It also does not require a final check that the metadata content was updated by the customer system.

Considering both test specifications, in order to implement corresponding tests, additional documents such as the interface Application Programming Interface (API) description for the CORBA- and the web service-interfaces are necessary.

4.3 Synchronization Tests

At the end of this section, the specification for tests that deal with the metadata exchange, referred to as synchronization in the standard, between two AEDP-17-conform servers should be considered [8, see L-2.7 Test Case “Synchronize with Another Server”]. In general, these tests describe the CORBA interface methods which must be invoked by a client, e.g., a synchronization service, in order to transfer products or associations between the systems.

Consider the following specification of a test for the synchronization of metadata between a supplementing system and the customer system [8, see L-2.7.1 Test Case “Synchronize metadata”]. The supplementing system, besides the customer system, is hereinafter referred to as the “reference system.”

L-2.7.1 Test Case “Synchronize metadata”

On the client library, where the client library is attempting to obtain catalog entries from a server library, perform the following sequence of steps to initiate a synchronization:

1. get Library instance from the server
2. get StandingQueryMgr instance
3. invoke CatalogMgr::submit_standing_query to retrieve entries with NSIL_CARD.dateTimeModified > t0, where t0 is time of last successful synchronization
4. invoke SubmitStandingQueryRequest::complete_DAG_results and wait for results
5. process query result set
 - (a) validate incoming DAG against supported data model
 - (b) update catalog entry for existing metadata OR
 - (c) insert new catalog entry for metadata
 - i. modify catalog entry NSIL_FILE.productURL and related file URLs in NSIL_RELATED_FILE.URL if applicable, to point to the client library
 - ii. modify NSIL_FILE.isProductLocal and NSIL_RELATED_FILE.isFileLocal if applicable, setting initial values to False
 - iii. store the original file and related file URLs for later retrieval (recommended)
6. record current time t1 as last successful synchronization time
7. cancel standing query that uses time t0
8. go to step 3. and repeat using time t1 for the standing query

The previous test case specifies that a permanent request should be performed by the client, e.g., a synchronization service. Thereby, new or updated metadata content

that has not yet been synchronized is retrieved from the reference system. After processing the metadata and updating some server-side attribute fields, the metadata content is stored in the customer system.

Tests that would be based only on this high-level specification have different disadvantages. The first one is that these tests would require that certain states or actions must be logged by the reference system, e.g., that a certain request was made on the reference system by the customer system. Testers then would have to consult these logs during a test and would have to manually mark these as failed or passed based on the logs. As another example, the synchronization between the customer system and the reference system has to be manually stopped during the execution of a test. This is necessary to prevent cheating, e.g., that the customer system retrieves a product again from the reference system instead of actually caching this product as requested by the standard.

Taking into account all of these limitations of the test case “Synchronize metadata” of the A.C.T.S., additional requirements regarding the capability to test the synchronization process have to be developed and implemented to achieve a correct and comprehensive test case.

5 Conclusion

In this chapter we analyzed how to use the test cases defined in the STANAG 4559 AEDP-17, to validate the standard conformity of systems. It was determined that the specification of the A.C.T.S. test cases serve as a starting point for the JISR Test Center to perform the basic interface tests of the AEDP-17. As the tests only provide a certain depth, it may be concluded that the AEDP-17 was primarily developed defining the communication between the systems and not to test the implemented interfaces. Passing all the A.C.T.S.-tests is necessary but not sufficient to achieve conformity to the STANAG 4559. As a result, the interface- and the synchronization-tests must be part of the JISR Test Center tests but have to be extended.

To be able to implement the A.C.T.S. tests for the JISR Test Center, the following extensions and definitions have been worked out:

- Test data to use in test cases (products and metadata) has been defined.
- The synchronization test specification has been analyzed and realized in concrete test cases.
- Manual steps, to be performed by the testers, have been automated wherever possible.

As the above-mentioned tests only provide an insufficiently low test depth to validate the standard conformity, additional testing capabilities for the JISR Test Center had to be evolved.

One major issue of the AEDP-17 specification is that the technical means are not linked to the actual use of the standard and thus many aspects are defined

only briefly. So in order to actually test relevant aspects, operational business rules and use cases needed to be identified. Such use cases follow the activities that are described in operational doctrine as provided within the JISR process. One document focusing on this topic is called MAJIIC 2 Business Rules and Use Cases (BRUC) and was created by the Architecture and Technical Working Group (ATWG) within the MAJIIC 2 (Multi-Intelligence All source Joint ISR Interoperability Coalition) [5] project. This document is about to become part of the STANAG 4559 as an additional AEDP and, thus, is relevant in this context. Although the BRUC does not contain any predefined test cases, the specified business cases and workflows served as the basis for considerations regarding further relevant test cases to increase the test coverage of the JISR Test Center.

After defining basic use case-based test cases, which are not considered in this chapter, and the described test cases for the AEDP-17, the JISR Test Center and the tests themselves have been implemented taking into account the requirements of Sects. 3 and 4. On behalf of the German Federal Ministry of Defence (German: BMVg), the JISR Test Center has been deployed at the WTD 81 in Greding, Germany and put into operation by the end of 2019 providing an initial operating capability (IOC). By now, only local testing is provided by the JISR Test Center. To provide easy access to the JISR Test Center and to extend the connection options for customers, remote testing should be supported as well. First tests to demonstrate remote testing capability were carried out successfully.

Now, the JISR Test Center must be made visible to the potential customers—the NATO nations and their partners—to establish the JISR Test Center as a central and independent testing facility for the STANAG 4559. Already developed business cases tests could be enhanced and expanded taking into account the feedback of the customers, e.g., concerning the relevance of test cases, as well as the experiences gained during testing, e.g., concerning test depth. In order to be able to provide test cases to cover the whole JISR Cycle, test cases facing the AEDP-18 and AEDP-19 could be added to the JISR Test Center at a later time.

Acknowledgments Parts of the work of the authors being described in this publication has been funded by the BMVg (Federal Ministry of Defence). The standard STANAG 4559 is developed within the CST of STANAG 4559. The authors acknowledge valuable help and contributions from all partners within the CST.

References

1. Coalition Warrior Interoperability Exercise (2020). <https://www.act.nato.int/cwix>. Accessed 06 February 2020
2. B. Essendorfer, C. Kerth, C. Zschke, Evolution of the coalition shared data concept in joint ISR. IST-SET-126 (2015)
3. B. Essendorfer, C. Kerth, C. Zschke, Adaptation of interoperability standards for cross domain usage, in *Next-Generation Analyst V*, vol. 10207 (International Society for Optics and Photonics, 2017), p. 102070E

4. D. Janzen, H. Saiedian, Test-driven development concepts, taxonomy, and future direction. *Computer* **38**(9), 43–50 (2005)
5. Lars Nesse, MAJIC Multi-sensor Aerospace-ground Joint ISR Interoperability Coalition (2006). <http://www.nato.int/docu/update/2007/pdf/majic.pdf>. Accessed 02 March 2020
6. NATO Standardization Office (NSO), AJP 2.7 Allied Joint Doctrine for Joint Intelligence, Surveillance and Reconnaissance (2016). <https://nso.nato.int/nso/nsdd/APdetails.html?APNo=1952&LA=EN>. Accessed 04 March 2020
7. NATO Standardization Office (NSO), AIntP-14 Joint Intelligence, Surveillance and Reconnaissance (JISR) Procedures in Support of NATO Operations (2016). <https://nso.nato.int/nso/nsdd/APdetails.html?APNo=2119&LA=EN>. Accessed 04 March 2020
8. NATO Standardization Office (NSO), NATO Standard ISR Library Interface-AEDP-17 (2018). <https://nso.nato.int/nso/nsdd/apdetails.html?APNo=2272>. Accessed 07 February 2020
9. NATO Standardization Office (NSO), NATO Standard ISR Streaming Services-AEDP-18 (2018). <https://nso.nato.int/nso/nsdd/apdetails.html?APNo=2273>. Accessed 07 February 2020
10. NATO Standardization Office (NSO), NATO Standard ISR Workflow Architecture-AEDP-19 (2018). <https://nso.nato.int/nso/nsdd/apdetails.html?APNo=2274>. Accessed 07 February 2020
11. NATO Standardization Office (NSO), STANAG 4559 - NATO Standard ISR Library Interfaces and Services (2018). <https://nso.nato.int/nso/zPublic/stanags/CURRENT/4559EFed04.pdf>. Accessed 07 February 2020
12. NATO Standardization Office (NSO). Stanrec 4777 NATO Intelligence, Surveillance, and Reconnaissance Interoperability Architecture (2018). <https://nso.nato.int/nso/nsdd/stanrecdetails.html?idCover=8591&LA=EN>. Accessed 07 February 2020
13. NATO Standardization Office (NSO), AIntP-14 Intelligence Requirement Management and Collection Management (2018). <https://nso.nato.int/nso/nsdd/APdetails.html?APNo=2254&LA=EN>. Accessed 04 March 2020
14. M. Shahin, M. Ali Babar, L. Zhu, Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices. *IEEE Access* **5**, 3909–3943 (2017)

Overall Scheduling Requirements for Scheduling Synthesis in Automotive Cooperative Development



Arthur Strasser, Christoph Knieke, and Andreas Rausch

1 Introduction

The development of embedded real-time systems in the automotive industry is cooperatively organized. Generally, the car manufacturer is responsible for system development, and suppliers are responsible for subsystem development. Since suppliers are specialized in designing and realizing subsystems (often for different manufacturers), they can develop subsystems from reused components (e.g., software components) in order to develop similar subsystems and to offer them cost-effectively to manufacturers [1].

The methodology defined by the automotive domain-specific AUTOSAR standard [2] contains the model-based development steps, *system development*, *subsystem development*, and *integration*. These steps are also applied in the cooperative component-based development. In the system development step, a coarse-grained decomposition of the software components, the hardware components, and their allocation are defined by using the AUTOSAR system description template. In the subsystem development step, the decomposition is refined into AUTOSAR atomic software components of the application layer, and the basic software layer is defined and implemented. The basic software layer is the hardware abstraction, which has to be realized by software components belonging to a certain hardware platform (e.g., real-time operating system). In the integration step, the execution environment, called AUTOSAR Run Time Environment (RTE), of the software components is configured using the ECU configuration description [3]. The configuration

A. Strasser (✉) · C. Knieke · A. Rausch
Clausthal University of Technology, Institute for Software and Systems Engineering,
Clausthal-Zellerfeld, Germany
e-mail: arthur.strasser@tu-clausthal.de; christoph.knieke@tu-clausthal.de;
andreas.rausch@tu-clausthal.de

contains the definition of the static scheduling. The execution order of executable software components is part of the static scheduling and must satisfy functional requirements, as defined in [4]. In general, the static scheduling is configured at compile time. The configuration is part of a configuration description, which defines a set of operating system tasks as well as processes triggered in a specific order by tasks for the execution of software components. In the following, we define the *partial scheduling* as the configuration of the execution order of AUTOSAR atomic software components belonging to an implemented software subsystem. Additionally, we define the *overall scheduling* as the execution order of atomic software components from subsystems that are integrated into the software system.

The software system has to satisfy functional and temporal requirements [4]. In this contribution, we focus on the description of functional requirements and in particular the interaction of subsystems. In AUTOSAR, the overall scheduling requirements are described in the AUTOSAR Timing Extensions (TIMEX) model. The syntax of TIMEX is similar to the structure of a totally ordered precedence graph that contains executable entities, depicted as nodes, and their precedence order, depicted as edges.

However, in the cooperative development, the system developer is not able to define the overall scheduling requirements using TIMEX because the decomposition of subsystems into executable software components is determined in later development steps by subsystem developers. Thus, the system developer can determine the overall scheduling only after the subsystem development step when integrating the subsystem models into the system model. To describe an overall scheduling, the developer determine an interleaving of the partial schedulings and if necessary modifies each partial scheduling. Two problems may arise in this step: First, there may be no interleaving, which satisfies the overall scheduling requirements. Second, an interleaving can be determined, but the dependability of the subsystems can no longer be guaranteed as the partial schedulings had to be modified for the overall scheduling.

To avoid these problems, the system developer must be able to describe the overall scheduling requirements on abstract architecture entities. The subsystem development then must comply with these scheduling requirements when implementing the partial schedulings.

Thus, we introduce an extension of the AUTOSAR standard, the so-called *PortChain* notation, in Sect. 4 to describe the intended interaction of subsystems. This description constitutes the overall scheduling requirements in our approach and is the basis for the later subsystem development steps. We also demonstrate by an example, how the *PortChain* notation is applied in our approach to enable the generation of an overall scheduling by the system developer.

The chapter is organized as follows: Sect. 2 gives an overview on the related work. The problem we address is motivated by an example in Sect. 3. Section 4 presents our approach to generate an overall scheduling from partial schedulings. Finally, Sect. 5 gives a conclusion and an outlook on future work.

2 Related Work

2.1 *Generating an Overall Scheduling*

In [5], Czarnecki classifies the generation of instances for concepts as the techniques *composition* and *transformation*. The composition technique is used to generate a component instance from a configuration description. The consistency of the description is verified by constraints. The transformation technique is used to generate a component instance by a set of transformations.

An example for generating instances is the AUTOSAR configuration generation step to instantiate the configuration description [6]. In order to realize the generation step from the AUTOSAR methodology, research and industry approaches make use of the MDA concept [7]. However, an overall scheduling has to be generated from instances of several configuration descriptions from different subsystem developers.

One example approach is based on the global scheduling heuristic from Scheickl [8]. Thereby, the schedulings of subsystems are merged into one global execution model. A constraint logic solver verifies the consistency of the merged set. The solver checks if the scheduling requirements are satisfied. In [9], Wozniak determines a global execution order from an initial scheduling configuration. This configuration contains calculated execution orders of runnable entities satisfying scheduling constraints. In a further step, a genetic algorithm-based heuristic is optimizing the initial configuration to target certain timing budgets. The system LET concept, which has been introduced in [10] to provide a model on the execution platform level using shared memory read–write instants and tasks, is another promising idea to determine a global scheduling.

2.2 *Modeling Automotive Embedded Systems*

Broy et al. define a set of architecture viewpoints and concepts for modeling vehicle systems. In particular, its common subset defines the functional layer, the logical layer, and the technical layer. These standard layers are the basis for developers of embedded systems modeling languages in the automotive industry [11].

The AUTOSAR domain-specific language (DSL) from the AUTOSAR methodology is an example for the realization of the technical layer concepts and the de facto standard in the automotive industry. The AUTOSAR DSL contains the TIMEX metamodel for modeling scheduling requirements. The consistency of TIMEX models determines whether the system scheduling model can be correctly realized. It is verified using the requirements and guarantees event chain constraints.

As another example, the EAST-ADL2 modeling language is realizing the concepts of the functional layer and the logical layer [12]. It is also used for modeling scheduling requirements. Therefore, it defines the TADL metamodel. The TADL syntax is similar to the event chain syntax of AUTOSARs TIMEX metamodel.

However, these interfaces cannot be used for modeling the overall scheduling requirements in the system structure design step of the system development as in cooperative component-based development software components are determined by a certain subsystem implementation.

2.3 Repository Organization for Model Artifacts

System developer and subsystem developer have to organize the exchange of model artifacts which can be metamodel instances based on different DSLs. Evans introduces its repository approach as a concept to enable the organization for object relations of different domain models during their life cycle. It encapsulates from their domain specifics [13].

The *GeneralStore* approach from Glaser et al. [14] is a well-known example realizing the repository idea. It targets the challenges of the cooperative development by providing a tool integration platform. The platform is built on the UML metamodel technology storing and organizing object relations by model to model transformations.

3 AUTOSAR Example: Display Controller

We introduce a simple demonstration example—a running light control application—to illustrate the approach. The letters R E D shall appear on a repeating display. For the display of one of these letters, a respective control signal is used. The order in which these signals occur determines the order in which the letters appear on the display.

3.1 System Structure and Informal Requirements

Figure 1 shows the system structure in accordance with the functional requirements. It contains the display component, the display controller software component, and

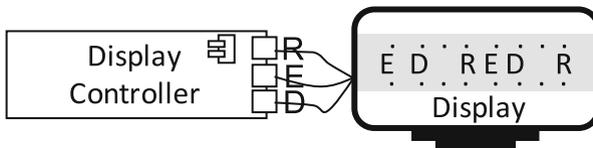


Fig. 1 System structure for the running light controller application

the connections between these components. Each time the controller is executed, it has to calculate a letter control signal. A letter signal is sent via the software component ports to the display. Then, one appropriate letter becomes visible after the other on the display. Three execution steps have to be triggered to show the letters R E D in that sequence. The execution steps are triggered cyclically.

Each calculation for the control signal must be ordered for execution in a way that satisfies the appropriate letter calculation and thus satisfies the required R E D signal sequence at the output ports. The top left corner of Fig. 2 depicts the decomposition of the controller software system in the AUTOSAR notation, which we call the *static structure* in the following. The static structure contains the subsystems BC and A. These subsystems are connected by abstract ports for internal

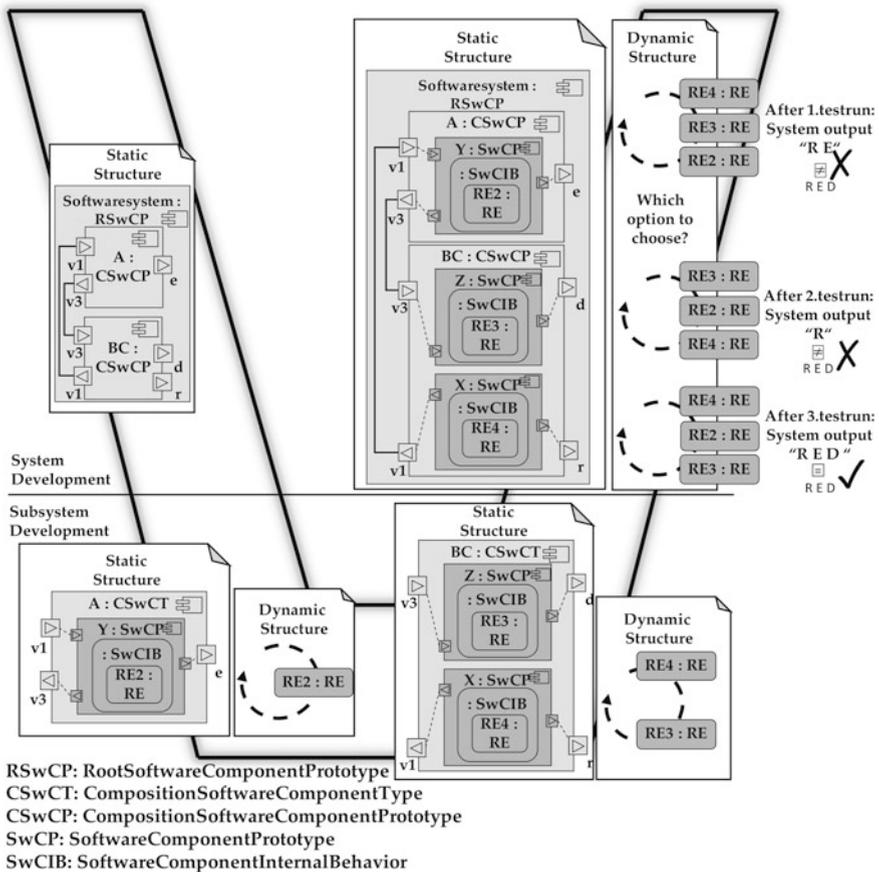


Fig. 2 Relevant development artifacts of the development steps in accordance with the V-model. The challenge in the integration step: finding a correct overall scheduling for the display controller example

communication. For each letter control signal, a port is defined. The abbreviations of the AUTOSAR types used by the AUTOSAR notation are introduced on the bottom left side in Fig. 2. AUTOSAR TIMEX requires the definition of executable software components. But the system development step in cooperative component-based development does not define these components. Therefore, the overall scheduling requirements can only be defined informally in cooperative component-based development. The following requirements as a sequence of abstract subsystem in and out ports exist to fulfill the desired R E D letter sequence:

- BC: (1. out r); (2. out v1);
- A: (3. in v1); (4. out e); (5. out v3);
- BC: (6. in v3); (7. out d).

3.2 *Subsystems and Partial Scheduling*

The decomposition into AUTOSAR atomic software components for the implementation of each subsystem is shown in the bottom of Fig. 2. We assume each subsystem to be developed independently by different suppliers. Subsystem BC contains the software components Z and X and the *runnable entities* RE3 and RE4. Subsystem A contains the software component Y and the runnable entity RE2. The runnable entities implement the behavior of the AUTOSAR *SenderReceiverInterfaces* defined by the in and out ports. They are triggered to execute the AUTOSAR atomic software component. The trigger is defined by the partial scheduling in the dynamic structure as depicted in Fig. 2: The dashed circle represents the cyclical execution of runnable entities. This is well known as the task of an operating system. Each runnable entity has a certain position in the order of execution represented as RE between the starting point and end point of the circle. The partial scheduling in subsystem BC is configured that RE3 is cyclically executed after RE4. In subsystem A, the partial scheduling defines a cyclical execution of RE2.

3.3 *Overall Scheduling as Interleaving of Scheduling*

The overall scheduling is created from the subsystems dynamic structure contents. Thereby, the execution orders are combined to a common cyclic execution order. As shown in the top right part in Fig. 2 (dynamic structure), several attempts are necessary to determine the final interleaving result for the desired R E D control signal sequence. The final overall scheduling of the example is RE4, RE2, RE3. RE2 has been interleaved between RE4 and RE3. Hence, the partial scheduling of subsystem BC has been modified.

4 Approach

As the subsystem BC has been modified, it is not clear, if all subsystem-specific requirements are still fulfilled. Hence, the system developer is not able to answer the question if the detected overall scheduling is correct in that scene, i.e., further quality assurance measures are required in this case.

Our approach aims to minimize this effort in the integration step by automation. Therefore, our approach introduces the *PortChain* description. It allows the system developer to describe the overall scheduling requirements. Each subsystem developer has to realize a partial scheduling in a way that these requirements are satisfied. If a correct overall scheduling exists, then the system developer is able to generate it from the given artifacts. The automation has to perform the following two operations: At first, the contents of the subsystem dynamic structures are combined, not modified, to one structure. Second, the missing connections between runnable entities of one partial scheduling and the others are derived for an overall scheduling.

4.1 *PortChain Description*

A *PortChain* describes how the subsystems interact over ports and connections with one another. The informal requirements from the example are represented formally by the *PortChain* notations in the top left of Fig. 3. A directed dashed circle specifies the order for the calculations of the control signals. These control signals are represented by the subsystems port notation. The ports are abstract since they abstract from the AUTOSAR interfaces. Thus, we introduce another port notation that is representing “W” port as an output port and an “R” port as an input port. The triggers for calculations from the *PortChain* are called *write port access* and *read port access*.

4.2 *Mapping Description*

For the derivation of an overall scheduling, the relation between the partial schedulings has to be known. Therefore, the subsystem developers describe how the subsystem satisfies the requirements from the *PortChain*. This description contains the mapping between runnable entities, port accesses, and the appropriate partial scheduling.

In the bottom right corner in Fig. 3, it is shown that RE4 is mapped to v1, r and RE3 is mapped to v3, d. The directed connections of port accesses define how the communication path along subsystems ports is ordered. As port access v1 is mapped to RE4 and port access v3 is mapped to RE3, the execution order of RE4 and RE3 must also satisfy the order of the port accesses: The execution order RE3→RE4,

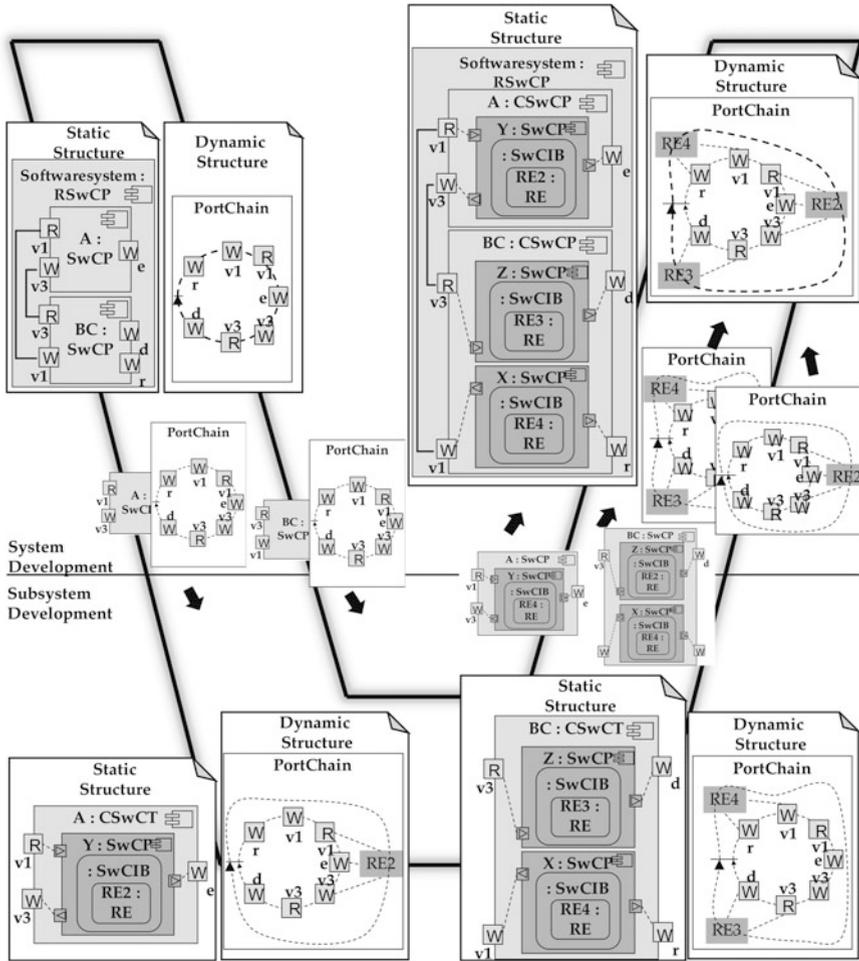


Fig. 3 Applying the approach for a model-driven generation of an overall scheduling

e.g., does not conform to the order $v1 \rightarrow v3$ as defined in the *PortChain*. A proper integration of subsystem BC must ensure that other port accesses of the *PortChain* that are left unmapped are also satisfied.

4.3 Overall Scheduling Derivation

To integrate the partial schedulings, all port accesses have to be mapped to runnable entities. The step is performed by our automation step. The input and output of the step are illustrated in the upper right corner of Fig. 3.

The mapping description of subsystem BC has the missing mappings of subsystem A and vice versa. Thus, the port accesses of both are overlapped to get a description that contains all mappings. The second operation determines the overall execution order in a way that the order of all port accesses is satisfied. In our example, the directed connections between RE3, RE2 and between RE2, RE4 from the *PortChain* have been derived. RE4 is executed first, followed by RE2 and RE3. This overall scheduling satisfies the port access order given by the *PortChain*. As the dynamic structure of subsystem BC was build satisfying the *PortChain*, the interleaving with RE2 does not lead to a modification that would require additional quality assurance.

5 Conclusion and Future Work

We proposed an approach enabling the specification of scheduling requirements by the system developer. Therefore, an extension of the AUTOSAR notation called *PortChain* was introduced. A *PortChain* describes how the subsystems interact over ports and connections with one another. Instead of describing the scheduling by the systems software component execution order, the communication sequence via ports is specified. When the software components are developed by the subsystem developers, the *runnable entities* of the components are mapped to the ports within the *PortChain* description to enable the generation of an overall scheduling. Thus, the integration task of finding a correct overall scheduling is facilitated significantly.

As a future work, we aim at extending the proposed methodology by taking temporal requirements into account. The next step is a prototypical implementation of the approach including the definition of an overall metamodel and a real-world case study.

References

1. M. Broy, I.H. Kruger, A. Pretschner, C. Salzmann, Engineering automotive software. Proc. IEEE **95**(2), 356–373 (2007). IEEE. <https://doi.org/10.1109/JPROC.2006.888386>
2. AUTOSAR Classic 4.4.0. <https://www.autosar.org/standards/classic-platform/>
3. AUTOSAR Classic 4.4.0 Methodology and Templates. <https://www.autosar.org/standards/classic-platform/>
4. H. Kopetz, *Real-Time Systems: Design Principles for Distributed Embedded Applications* (Springer, New York, 2011)
5. K. Czarnecki, *Generative Programming: Principles and Techniques of Software Engineering Based on Automated Configuration and Fragment-Based Component Models* (Ilmenau University, Ilmenau, 1999). <http://d-nb.info/958706700>
6. R. Hebig, Methodology and Templates in AUTOSAR. Potsdam University, Technical report (2009)
7. MDA Guide revision 2. <https://www.omg.org/>

8. O. Scheickl, *Timing Constraints in Distributed Development of Automotive Real-time Systems* (München University, Munich, 2011)
9. E. Woźniak, *Model-based Synthesis of Distributed Real-time Automotive Architectures* (Université Paris-Sud, Orsay, 2014)
10. R. Ernst, L. Ahrendts, K. Gemlau, System level LET: mastering cause-effect chains in distributed systems, in *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, (2018), pp. 4084–4089. <https://doi.org/10.1109/IECON.2018.8591550>
11. M. Broy, M. Gleirscher, S. Merenda, D. Wild, P. Kluge, W. Krenzer, Toward a holistic and standardized automotive architecture description, in *Computer*, vol. 42(12) (2009), pp. 98–101. <https://doi.org/10.1109/MC.2009.413>
12. P. Cuenot, P. Frey, R. Johansson, H. Lönn, M. Reiser, D. Servat, R.T. Koligari, D. Chen, Developing automotive products using the EAST-ADL2, an AUTOSAR compliant architecture description language, in *Proceedings of the 4th European Congress ERTS (EMBEDDED REAL TIME SOFTWARE) SIA* (Société des Ingénieurs de l'Automobile - French Automotive Engineers Society. (2008). URN: urn:nbn:se:kth:diva-81221
13. E. Evans, *Domain-Driven Design: Tackling Complexity in the Heart of Software* (Addison-Wesley Professional, Boston, 2004)
14. K.D. Muller-Glaser, C. Reichmann, P. Graf, M. Kuhl, K. Ritter, Heterogeneous modeling for automotive electronic control units using a CASE-tool integration platform, in *IEEE International Conference on Robotics and Automation*, New Orleans, LA (2004), pp. 83–88. <https://doi.org/10.1109/CACSD.2004.1393855>

Extracting Supplementary Requirements for Energy Flexibility Marketplace



Tommi Aihkisalo, Kristiina Valtanen, and Klaus Käsälä

1 Introduction

Requirements engineering tackles the demanding task of extracting requirements out of the perceived properties of the software system in design phase with varied methodology. This process is usually impacted with the factors natural to the human communication and arbitrary nature of it. This paper presents extraction, results, and analysis of requirements sourced from the expert panel's discussion notes. The requirements are considered supplementary for an existing and running early version of the proposed Internet-based flexibility market exchange FLEXIMAR service [1]. The FLEXIMAR market is two-sided single-tier double auction type of *market institution* in sense of [2].

The flexibility or demand response in electricity usage manifests typically as time shifted or time scaled energy usage as defined by the IEA [3]. On the FLEXIMAR market, the flexibility is handled as a tradeable commodity cleared in continuous trading exchange. It is intended for trading even small batches of electrical energy consumption flexibility on a single tier for sourcing the flexibility even from the household level as much as possible. The main goal was to aim for markets with a highly valued short response time and high volatility flexibility. This value can be found on reserve markets [4] requiring harnessing smaller and more compact but fast and agile resources. To have these resources available continuously with known location and availability, latency time would greatly improve the stability and reliability of power grids.

As FLEXIMAR is mainly intended to harvest and utilize even the small parcels of flexibility on the household level, the trading transaction costs must be low. The low-

T. Aihkisalo (✉) · K. Valtanen · K. Käsälä
VTT Technical Research Center of Finland, Oulu, Finland
e-mail: tommi.aihkisalo@vtt.fi; kristiina.valtanen@vtt.fi; klaus.kansala@vtt.fi

cost trading with a low value is made possible by technical solutions automating the exchange trading and respective premises' equipment control. This would be, e.g., smart-meter based or add-on system for allowing unattended equipment control and the flexibility trading.

The bare bone marketplace with core functionalities was already implemented in minimum viable product creation fashion without a traditional and extensive design phase. The purpose was to produce early a working exchange environment for prototyping and trialing the various connected trader and environment control systems in practice. The minimum functionalities were created, but clearly it was noted that there are several more or less energy domain and trading-related open questions requiring more detailed attention and expertise than there was available at that moment.

Therefore, a workshop overseen by the authors was held to gather and utilize the project consortium's expertise to supplement and improve the existing marketplace system. The expert panel had already been somewhat familiar with the basic design choices and other details of the marketplace but the comments or not alone the requirements were never gathered before. The further details of the system were presented to the workshop audience in order to stimulate the discussion. The two groups formed out of the participants were recording their findings and discussion on the worksheets and on which the research in this paper is based. These discussions and their recorded results were discovered to reflect various supplementary system requirements, functional or nonfunctional, directly or suggestively for the existing bare bone FLEXIMAR platform. After the analysis, results can be utilized as ready requirements and requirement suggestions that has to be finally determined at least in the later marketplace service and product creation phase. These are valuable already for the further improvement and development of the FLEXIMAR flexibility marketplace.

The contribution of this paper is the collection of the supplementary requirements but also the set of requirement suggestions and their respective analysis for the energy flexibility marketplace service enabling direct consumer-based trading.

2 Related Work

The FLEXIMAR marketplace is considered as a single-tier marketplace unlike [5] in where market brokers may participate to the secondary or other markets too. However, multilevel participation by the traders is not prohibited on FLEXIMAR. It may be even considered natural mechanism for the aggregator style of participants sourcing the flexibility from the lower tier. These business models are covered in [6] as direct trade and agented trade. The direct trade is suitable for those willing to assume the financial risk due to it [6] as it is the case here.

The traders are expected to be responsible to consider all factors impacting their justified price determination. The FLEXIMAR properties of the commodity include at the moment only the offered volume and timing. The traders may be

interested in, e.g., the source’s location information for loss and pricing estimation [5], type of flexibility source, etc. In this open market environment, the traders are considered equal, and no priorities are given, e.g., to distribution system operators (DSOs) or transmission system operators (TSOs) like in [7]. Reference [8] lists further the different options of DSO-TSO and DSO-DSO co-operations on the markets with respect to other marketplace solutions like FLEXIMAR. More relevant considerations of the same reference include operational responsibility, integration to other markets, reservation payments, and commodity standardization showing deviation between the listed solutions.

Plenty of research exist regarding the optimal bidding, possible types of market mechanisms to realize pricing, and the auction resolving mechanisms optimizing toward varied outcomes in various market and power network settings. Just to name a few considering two-sided auction are, e.g., [9–11], providing simulated analyses but not the system-level requirements.

3 Architecture and the Expert Panel

3.1 Overview on the FLEXIMAR Architecture

The FLEXIMAR market is an Internet-based service constructed with ready components or COTS as far as possible allowing high-speed trading transactions. The trading environment consists of the main marketplace and a number of connected client flexibility trading systems interacting with it as presented in Fig. 1.

This high level architecture regarding particularly the client side is only suggestive leaving the internal design to the trader system implementer. The flexibility trader’s automated trading agent communicates through the AMQP [12] trading messaging interface for submitting the desired trade orders and receive related communication back from the marketplace. The equipment control part is responsible of controlling the supply or demand of the energy flexibility in the trader’s environment which may include simple heating or lighting system or a full-blown industrial machinery environment. The other exposed interfaces are the WWW

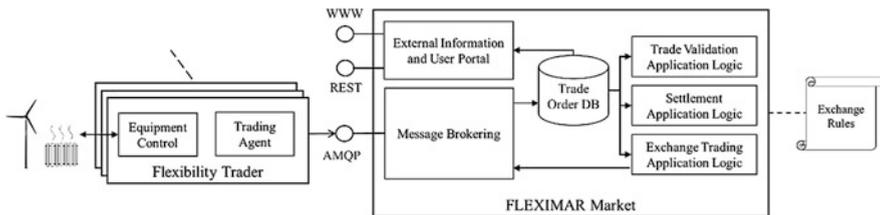


Fig. 1 The main functional entities including the client side flexibility trader and the FLEXIMAR market with their respective internal main components

interface for the human readable market portal and the REST-based interface for machine-readable market information. The entity serving these interfaces handles also the user account and access authorization management tasks. During the user registration process, the trading user is required to commit to the exchange rules dictating the preconditions, conventions, and penalties on the exchange. In practice the rules are enforced by the message processing and application logics.

The other internal major parts of the exchange include message processing the in- and outgoing messages with brokering, queuing, and content validation functionalities. The messages are relayed to the trade order database which is a time series database to store and manage the incoming trade order messages. The stored trade order messages are matched in FIFO-order by the micro-service-based exchange trading application logic implementing a modified auction market mechanism as defined by the exchange rules. In this, the highest bidder will be obliged to purchase the complete seller's lot provided that the flexibility's time and energy allocations do overlap at least partially. The optimality and effects of this mechanism are not analyzed here and will be left for the future. The transactions covered in the executed and cleared trades is expected to happen as agreed and which are validated by the trade validation application logic based on the certified metered values and trading logs. The respective rule defined penalties will occur if discrepancies are recorded.

Finally, the settlement application logic will manage the settlement between the involved parties of the cleared and validated individual trades to transfer the funds. In order to minimize transaction costs and conceivably improve system security, block chain or more widely distributed ledger technologies (DLTs) were identified to have various capabilities resonating with system requirements of the settlement process [13]. However, considering the maturing state of the DLTs, regarding, e.g., real-time performance [14, 15], in the FLEXIMAR platform, the role of the technology is limited to the settlement process in which timing requirements are not so stringent. The settlement process itself has the most open questions, and care is required due to dealing with the actual transfer of funds between the participants on the marketplace.

This architecture and its main functionalities and entities formed the guiding framework for the expert panel's discussion and notes for the further analysis in this work.

3.2 The Expert Panel

The workshop was held in Espoo, Finland, in December 2019 with FLEXIMAR project partners and other related stakeholder representatives. The workshop participants represented various project partners totaling to 18. Represented were 3 separate energy system ICT solution providers and consultants by 5 participants, national transmission grid operator by 1 participant, data protection and security tool and solution provider by 1 participant, and 2 research partners by 11 with strong ICT,

electricity system, and market research background. Two groups consisting of nine members each were formed to discuss and record their findings on the worksheets. The worksheets were intended to record views on and guide the discussion to several categorical topic areas of the system. These categories consisted of the nature and possible requirements of the settlement process (SP), user certification process (UC), penalties (P), threats and misuses (TM), terms of service (TS), and the other (O) for miscellaneous.

4 Results

In total nine sheets were collected from the panels. The recorded results were quite freely formed notes reflecting an individual expertise and views on the matters discussed. The worksheets' structure was not seemingly followed very carefully, resulting in requiring more analysis effort. The worksheet outputs in style can be divided into question and statement outputs regarding the technical and other procedural conventions. Clearly, the question type outputs set a requirement suggestion to be considered, while the statements set a requirement more directly. The requirement suggestions can be formed to the actual requirements later after renewed consideration by the expert groups or product creation and business decision-making process. Even as such, they are valuable pointing out the system's varied aspects that require attention.

In order to capture the actual requirements, the contents of the filled worksheets were read through and then recapped. The resulting initial output after this is presented in Table 1 . The recapping included generalization with an attempt to preserve actual contents but also combining the multiple outputs meaning basically the same. The number of appearances in different forms is indicated also. The suggestive requirements are marked with a question mark. Worksheet categories as explained in the previous chapter are reflected by the ID.

Terms of service (TS) target category did not catch any outputs.

5 Analysis

For the analysis, the results in Table 1 were rearranged once more to form more descriptive and uniform topic categories. The original placement into the subtopics by the work groups did not always make sense as the worksheet output was free formed anyway. The new subtopic groups defined heuristically are Settlement (S), Market Participation (MP) and Market Operating (MO). The Settlement topic group covers all items related to the settlement process, Market Participation about the rules and conditions regarding the participation to the market. Market Operating includes mainly market governance and operative issues.

Table 1 Worksheets outputs ID'd, combined, and recapped with a number of appearances

ID	Content	Appearances
SP1	Currency used in trading and its exchange rate?	2
SP2	Size of the payments in the settlement?	1
SP3	Frequency of the payments in the settlement?	2
SP4	Verification of the settlement balance?	1
SP5	Taxation of the resulting trading profit?	1
SP6	Settled balance usage options (toward trading, only payout)?	1
SP7	Minimum and maximum size of the transactions?	1
SP8	Types of the allowed participants on the marketplace?	1
SP9	Pricing alternatives on the marketplace?	1
SP10	Coordination with other markets?	1
SP11	Threshold to payouts in the settlements?	1
SP12	Deposit on the account needed to buy, but also to sell?	1
SP13	P2P trading possibility?	1
SP14	Metering equipment and metering frequency?	1
SP15	Validation of flexibility flow?	3
SP16	Settlement through metering value adjustment?	2
SP17	Role of and information feed to the balance responsible parties?	1
UC1	User identified and recognized by the national or third-party client registry	5
UC2	User's flexibility and delivery capacity negotiation?	2
UC3	User certification process for the aggregators or other non-prosuming users?	2
UC4	Simplicity in the rules and participation models needed	1
UC5	Account deposit required to participate?	1
UC6	User's account must be sufficiently funded to allow participation	1
UC7	User prequalification required to determine allowed trade volumes?	2
UC8	User certification in phases in order of the participant size?	1
UC9	Technical user equipment certification?	1
P1	Responsibility for malfunctioning market platform?	1
P2	Bank guarantee or smart contract-based process to mitigate misuses?	2
P3	User account deposit lost if misused detected	1
P4	Not delivered flexibility capacity should lead to penalty	2
P5	Network balancing responsibility?	1
TM1	Prevention or minimization of gaming on the market?	3
TM2	Aggregators or other non-prosuming users allowed?	1
TM3	Platform failures and wrong operation?	1
O1	Marketplace data ownership and protection?	1
O2	Liabilities and responsibilities for technical failures?	1
O3	User equipment costs?	1
O4	Aggregation type of usage should be possible	1
O5	Location information for the user's trading metering points?	1
O6	Trading yields toward balance settlement at the metering points?	1
O7	User account limits on trading to mitigate effects of accidental trades?	1
O8	Balance responsibility if balances moves up due to the flexibility trades?	1

Table 2 presents these generalized new requirement outputs in the rearranged subtopics in the descending number of appearances for the further analysis about the assumed importance. In the Type column, “W” indicates a requirement suggestion to be considered, while “R” indicates a more explicit requirement. The judgment of the placement to either type was based, besides the output content, on the authors’ understanding of the existing system’s state of development and definition. The Related column refers to initial requirements that appear in Table 1 for each new generalized requirement.

All two highest number of appearances are located in the Market Participation category. In this MP subcategory, the first four outlines the requirements for the participators’ prerequisites and terms to participate to the marketplace. Clearly, the requirement for the clients’ recognition and identification is clear. However, the

Table 2 Generalized requirements in each topic category, its type, number of appearances, and related source requirement in the previous table

ID	Type	Requirement	#App.	Rel.
S1	R	Not delivered or consumed flexibility or other failure leads to penalty toward balance/deposit	5	P2, P3, P4
S2	W	Transaction and settlement sizes, threshold, min/max	4	SP2, SP4, SP7, SP11
S3	W	Currency or other type of tender used in the settlement	3	SP1, SP6,
S4	W	Alternative use of settlement balance, e.g., toward metered balance	3	SP16, O6
S5	R	The trade transaction executions are validated and verified	3	SP15
S6	W	Settlement payout frequency	2	SP3
S7	W	Taxation of the settlement yields	1	S7
MP1	R	User identified, localized, and recognized by the national or third-party client registry	6	UC1, O5
MP2	R	Deposit/guarantee on the user account needed to cover volumes of trading	6	SP12, UC5, UC6, UC7, O7
MP3	R	User’s equipment, flexibility, and delivery capacity certification	5	UC2, UC8, UC9, SP14
MP4	W	Other than prosumer client types allowed and respective certification	5	SP8, UC3, TM2, O4
MP5	W	Balance responsibility	3	SP17, P5, O8
MP6	W	P2P trading possibility	1	SP13
MP7	W	Responsibility of client equipment costs	1	O3
MP8	R	Simplicity in rules and participation needed	1	UC4
MO1	W	Gaming the markets undesired	3	TM1
MO2	W	Operational responsibility of the marketplace	3	P1, TM3, O2
MO3	W	Coordination with other markets	1	SP10
MO4	W	Data ownership and protection	1	O1

intended solution for it in Finland is the DataHub, the national electricity customers' registry, but its deployment is being delayed, so other alternative solutions must be pursued instead. This would have allowed usage of nationally recognized registry of metering points and associated customers.

Next, the clients' market engagement should be limitable, based, e.g., on the clients' certified equipment capacity, MP3, and covering funds on the accounts, MP2. This allows setting a hard maximum for the acceptable trading volumes so that it must be always covered by the account balance. This requirement may be supplemented by the participant's certified technical flexibility capacity if available. The technical certification for the non-prosuming or retailing clients must be defined separately if allowed at all in the final solution, MP4. Such clients, e.g., aggregators, are participating only for trading without actual own flexibility consumption and/or generation capability. The harder financial limits setting may suffice for such clients but calls for a business decision. The size of the actual deposits or guarantees was not discussed.

In the Settlement subtopic requirement, suggestions are similarly calling for a limited size of the trades, S2, and its respective effects on the clients' account, S1. Accounts and their balances should be set in a currency, S3, but also should allow alternative uses for it, e.g., using it toward the metered electricity consumption charges, S4. The currency is not decided on, but it would be natural to at least represent it as local physical currency even if the actual transactions are settled in some technical block chain-based currencies.

The validation of the delivery of the flexibility or consumption changes according to the executed trades is required before the settlement, S5. In case of failure to deliver or consume, the penalties should apply toward the client's existing account balance including the deposit or other form of guarantee, S1. In order to validate the consumption changes, a change by volume and the time of it must be recorded at the trading partners metering points periodically.

Some less mentioned requirement suggestions, S6 and S7, are related to the frequency of the settlement process execution and payouts but also the taxation, which may end up to be inevitable anyway as the system deals with the monetary profits and losses. The clients' equipment costs in order to participate to the markets and how they are covered would require more business-level consideration, MP7. The same applies to the simplicity of rules and conditions to participate and trade to create an attractive marketplace and environment. Thus, they are not covered in this research.

In the Market Operation category, the market attractiveness-related requirements are calling for the prevention of gaming on the markets but also the operational responsibility of the marketplace service, MO1 and MO2. The responsibility question may be something that this research will be unable to answer as it is out of the scope for it. The same applies to the data ownership and its governance. However, such questions should be taken into account in the terms of service when the marketplace is open for the public. Coordination with other similar flexibility marketplaces, MO3, if technically and otherwise viable may help enhance liquidity of the traded flexibility in general.

6 Discussion

Clearly, the worksheet design must be such that it helps to focus on the topic in the form of guidance and clarity. The more carefully prepared pre-session introduction can also help. However, as the results in this approach may require more analysis and recapping, it enables also perhaps less restricted output and thus offer more insight on issues not obvious and imaginable for the system designers before.

The results basically indicate that this kind of marketplace should establish a secure and fair environment protecting the trading participants' interest within the individual and agreed limits. This concerns especially the trading rules and safeguards for the participants by limits of their trading accounts and equipment capacities but also the validation of the executed trades and the verification of all participants' identity.

This requires setting clear safeguarding limits for the client participation such that will protect his financial assets by limiting the trade to the value of the account liquidity and the technical limits of flexibility provisioning. The actual limits and sizes of the required deposits to open a trading account were not discussed as it may involve more business and service design before the actual public deployment. The technical flexibility limits may be established by the certification of the users and their equipment. Similarly, the participating clients' identity should be recognized and thus certified by the electricity consumer registry or something similar. These are the main thing the trading environment should facilitate as the responsibility of the actual trading is mainly left for the traders. At the same time, the trading-associated responsibilities and the associated rules should be understandable and justifiable. This includes outlining the costs of participation but also the required equipment before the public deployment of such service.

The trades on the marketplace and their execution, i.e., the respective modification of consumption or production at the trading partners, must be validated to maintain credibility and fairness of the marketplace. The validation is naturally achievable by metering the consumption at both parties and can take place only after the flexibility exchange has ended as agreed on the trade. Clear limitations and requirements for it were not found here as it also depends on the permitted future time frame for creating the trade orders.

Several requirement suggestions clearly should involve business decision and service design to take place. This would require some parties taking the responsibility of operating the marketplace service and making the related business decisions. Furthermore, the question of the data ownership and governance should be defined also, but in the end the data ownership may one of that factors impacting onto the desirability of operating such marketplace. One of the things to consider also at the same time is the desirability of permitting gaming or speculative trading on the market platform. Some of the work group's outputs call for prevention of such trading, but such limits may however even restrict the desirability to participate for some and thus limit the volumes on the platform. Also such prevention may call for reworking of the marketplace's ruleset among with the technical solutions.

Moreover, distributed ledger technologies, DLTs, can satisfy many supplementary requirements of the settlement process but also more general require-

ments related to the market participation and the market operation. The small or nonexistent transaction costs and straightforward integration to the real-world currencies widen the array of the viable flexibility business opportunities because many business-related design aspects may not be anymore so constrained by the technical limitations. In the future, the increasing role of digital currencies and the machine-to-machine economy, harnessed with DLTs, expands the possibilities of the FLEXIMAR platform and consumers' flexibility profits to be seamlessly integrated to other services, also outside energy sector.

References

1. VTT Technical Research Centre of Finland, FLEXIMAR - Novel marketplace for energy flexibility (2019), <http://www.fleximarex.com>. Accessed 2 Mar 2020
2. Friedman D, The double auction market institution: A survey, in *The Double Auction Market Institutions, Theories, and Evidence*, (1993), pp. 3–26. <https://doi.org/10.4324/9780429492532-2>
3. International Energy Agency and 21CPP, Status of power system transformation 2018 - Advanced power plant flexibility (2018). Available <https://webstore.iea.org/status-of-power-system-transformation-2018>
4. H. Hellman, A. Pihkala, M. Hyvärinen, et al., Benefits of battery energy storage system for system, market, and distribution network – Case Helsinki. *CIREC Open Access Proc. J.* **2017**, 1588–1592 (2017). <https://doi.org/10.1049/oap-cired.2017.0810>
5. D. Case, M. Nazif Faqiry, B. Majumder, et al., Implementation of a two-tier double auction for on-line power purchasing in the simulation of a distributed intelligent cyber-physical system. *Res. Comput. Sci.* **82**, 79–91 (2014). <https://doi.org/10.13053/rcs-82-1-7>
6. G. Luo, Y. He, C. Zhao, et al., Coordinated wholesale and retail market mechanism for providing demand-side flexibility. 2019 IEEE sustainable power and energy conference (ISPEC) (2019). <https://doi.org/10.1109/ispec48194.2019.8975245>
7. R. Fonteijn, T. Van Cuijk, P. Nguyen, et al., Flexibility for congestion management: A demonstration of a multi-mechanism approach. 2018 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe) (2018). <https://doi.org/10.1109/isgtteurope.2018.8571896>
8. T. Schittekatte, L. Meeus, Flexibility markets: Q&A with project pioneers. *Util. Policy* **63**, 101017 (2020). <https://doi.org/10.1016/j.jup.2020.101017>
9. M. Faqiry, S. Das, Double-sided energy auction in microgrid: Equilibrium under price anticipation. *IEEE Access* **4**, 3794–3805 (2016). <https://doi.org/10.1109/access.2016.2591912>
10. Z. Zhang, H. Tang, Q. Huang, W. Lee, Two-stages bidding strategies for residential microgrids based peer-to-peer energy trading. 2019 IEEE/IAS 55th industrial and commercial power systems technical conference (I&CPS) (2019). <https://doi.org/10.1109/icps.2019.8733335>
11. H. Mohsenian-Rad, Optimal demand bidding for time-shiftable loads. *IEEE Trans. Power Syst.* **30**, 939–951 (2015). <https://doi.org/10.1109/tpwrs.2014.2338735>
12. AMQP (2008), AMQP advanced message queuing protocol: Protocol specification version 0.9.1, Nov 2008
13. K. Valtanen, J. Backman, S. Yrjola, Blockchain-powered value creation in the 5G and smart grid use cases. *IEEE Access* **7**, 25690–25707 (2019). <https://doi.org/10.1109/access.2019.2900514>
14. R. Han, G. Shapiro, V. Gramoli, X. Xu, On the performance of distributed ledgers for Internet of Things. *Internet Things* **10**, 100087 (2019). <https://doi.org/10.1016/j.iot.2019.100087>
15. M. Andoni, V. Robu, D. Flynn, et al., Blockchain technology in the energy sector: A systematic review of challenges and opportunities. *Renew. Sustain. Energy Rev.* **100**, 143–174 (2019). <https://doi.org/10.1016/j.rser.2018.10.014>

A Dynamic Scaling Methodology for Improving Performance of Data-Intensive Systems



Nashmiah Alhamdawi and Yi Liu

1 Introduction

Currently, we live in an age where data has emerged and is drawing attention in several fields such as science, healthcare, business, finance, and society. The continuous growth in data size in various fields has caused an overwhelming flow of data in the last decade. Thus, many systems have faced problems analyzing, storing, and processing large quantities of data, which in turn has caused performance problems, such as slow responding time and bad usability [1]. When experiencing poor performance in systems processing huge amounts of data, a negative impact is created in increased costs, decreased revenue, or both. Additionally, poor performance causes delays, creates unprocessed data, and increases response time. In this paper, we especially focus on the slow responding time/slow processing speed as the performance issues.

With data-intensive systems becoming more prevalent, a need exists to overcome the challenges and implications of massive data. For example, the Epidemiological Applications of Spatial Technologies (EASTWeb) system [2, 3], which we have developed and used for early detection and early warning of mosquito-borne diseases, had suffered from severe performance problems due to the increasing volume of the datasets. EASTWeb automatically downloads earth observation datasets, processes them, generates, and stores the statistical summaries for a given

N. Alhamdawi

Department of Electrical Engineering and Computer Science, South Dakota State University, Brookings, SD, USA

Y. Liu (✉)

Department of Computer and Information Science, University of Massachusetts Dartmouth, Dartmouth, MA, USA

e-mail: yliu11@umassd.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_41

577

area and time period. The sizes of the earth observation datasets vary from 200 mega types to 1 giga types for a day. While processing a project with 30 years of data (1 giga types each day), EASTWeb significantly slowed down after processing 10 years of data and almost halted after processing 15 years of data.

The goal of this research aims to identify how to enhance the performance of data-intensive systems from the architectural perspective. We have developed a dynamic scaling methodology using virtualization to improve performance of data-intensive systems and to establish an approach to adapt such a system to the deployment environment. We have applied this methodology to the EASTWeb system and EASTWeb took 45% less time to process the same amount of data, and the projects did not halt and were finished within the expected timeframe.

This paper presents the dynamic scaling methodology for improving the performance of data-intensive systems. The outline of the remainder of the thesis is organized as follows. Section 2 summarizes the background of the research including the EASTWeb system and the cloud computing environment. Section 3 illustrates the details on the dynamic scaling methodology. Section 4 uses EASTWeb as a case study to show how to apply the dynamic scaling approach and demonstrates the results of the performance improvement. Section 5 discusses related works similar to our approach and Sect. 6 concludes the work.

2 Background

2.1 EASTWeb Application

The Epidemiological Applications of Spatial Technologies (EASTWeb) application [2, 3] is developed as an open-source, client-based application that automatically connects to earth science data archives. It helps acquire, process, and summarize remote sensing data according to the time period and geographic information provided by the user. All the information summarized is saved in a database that can easily inquire and connect to the data server, ecological and epidemiological, for further analyzation and prediction in a software environment. EASTWeb is implemented in Java, and utilized PostgreSQL to save and manipulate the resulting data summaries. The system inputs are a collection of earth observation data files which are already downloaded from online archives. The system outputs tables including data on the summary statistics of environment indices for each special zone. EASTWeb implements eight plugins for eight different earth observation datasets including GPMs IMERG, IMERG_RT, MODIS MOD11A2 C6, MODIS MCD43A4 C6, TRMM_3B42, TRMM_3B42RT, NLDAS forcing, and NLDAS NOAH. Each plugin has various indices ranging from 1 to 15 indices. The indices do various environmental calculations for each data product. The user can use EASTWeb to create a project that may include several plugins and choose indices associated with the selected plugins.

2.2 *Cloud Computing Environment*

Cloud computing is an emerging area, supporting various fields of computing, and has become a strong architecture to apply massive-scale and complex computing. Cloud computing facilitates in providing the virtualized resource, parallel processing, data storage, and security [4]. It is invented to enable a capable access to share resources, such as computer networks, servers, storage, and application services. Moreover, it assures to be less expensive compared to supercomputers and specialized clusters, is a more reliable platform alternative to grid, and is more scalable than clusters [5].

The cloud computing environment is provided by vendors such as IBM, Microsoft Azure [6], Google Cloud Platform [7], and Amazon Web Services [8]. Amazon Web Services (AWS) is utilized as the deployment environment of this study. AWS is a secure cloud service platform offering several functionalities helping businesses scale and grow.

In this research, we have used Elastic Compute Cloud Service (Amazon EC2), Cloud Storage Service (Amazon S3), Amazon Relational Database Service (Amazon RDS), and AWS CodeDeploy Service in the case study of our methodology.

Amazon Elastic Compute Cloud Service (Amazon EC2) is a web-based service allowing businesses to execute applications in the public cloud. Amazon EC2 allows a developer to create virtual machines (VM) known as instances, which can easily configure the capacity scaling of computing. Moreover, utilizing Amazon EC2 eliminates the needs of expensive hardware, so it provides the ability to develop and deploy applications faster [8]. Amazon Simple Storage Service (Amazon S3) [9] is a scalable, low-latency, affordable, web-based cloud storage service.

Amazon S3 facilitates online backup, archiving data, and storing applications. Also, it is designed to make web-scale computing easier for developers. With Amazon S3, data are stored in sealable containers known as buckets. A bucket is able to store several kinds of data and can be controlled and managed by the user.

The Amazon Relational Database Service (Amazon RDS) is a web-based service that facilitates setting up, operating, and scaling a relational database in the cloud [8]. Amazon RDS offers an affordable scaling capacity for an industry standard relational database. It is designed to manage database tasks such as backup, migration, and patching. Moreover, Amazon RDS supports six familiar database engines, including PostgreSQL, MySQL, Oracle, Amazon Aurora, MariaDB, and Microsoft SQL Server [8].

AWS CodeDeploy is a deployment service that automates an application to an Amazon EC2 instance and on-premise server. For successful deployment, a developer should define three criteria: revision, deployment group, and deployment configuration [10]. The revision is the content deployed onto instances such as code, web, and configuration file. Also, it is stored in S3, GitHub repositories, or Bitbucket repositories to be able to be deployed by AWS CodeDeploy. In the deployment group, a set of instances related to certain applications is specified by the developers.

Deployment configuration determines the steps of the deployment process to assure the revision is set at appropriate instances.

3 Methodology

We developed a novel dynamic scaling methodology to improve performance by applying an algorithm to scatter the system and scale up/down the system in the deployment environment. The methodology contains three steps: splitting and modifying the system whose performance needs improvement, choosing the deployment environment, and transforming the database.

3.1 *Scaling the System*

A system dealing with a large amount of data in a single computer can cause troublesome performance due to processing massive transactions at the same time in a single running space. In order to respond to this problem, our approach divides the massive amount of data into smaller pieces handled in parallel computers.

The helper project algorithm is developed to split the amount of data needed by the system to collect, process, and store, and it works based on several rules that prioritize dividing from high to low. The algorithm goes through these rules and starts splitting the data into small pieces to be processed in a timely manner. These rules consider every aspect when dividing the data into size of the data, type of data being collected, and kind of operation and processing applied on the data. Then, each aspect is checked to see if it requires splitting.

Rule 1: Consider Categories of Data in Splitting

This rule is a high priority to be checked first by the algorithm. The category of data processed implies how to capture, process, and store these data, and each category has its own individual way to do so. The helper project algorithm checks if the system processes several categories of data by finding the number of categories necessary to begin the process. Then, it starts splitting the data category based on the number of categories for each piece based on a category to be processed at a timely manner. For example, if the system has five different categories of data needed to be processed, the helper project algorithm splits the data into five small pieces. Then the system processes each small piece in parallel instead of processing all the categories at once. Typically, the system applies several operations to process and analyze the data to gain the desired result from these data. Consequently, the helper project algorithm checks the second rule for separation.

Rule 2: Consider Analyzing Data in Splitting

After the helper project algorithm examines and applies the first rule and does the necessary separation, it checks how to analyze the data. This rule regards any process of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information [11] or any operation applied on collecting data to gain desired results. Therefore, the helper project algorithm demands to find how many operations or events are applied on the collected data; based on this number, a decision is made on how to split the analysis of the data. The maximum number of operations that applies on data for separation varies from system to system. To determine the maximum number of operations that the system is able to tolerate, it conducts a sequence of tests on the system with a different number of operations applied. Based on the maximum number of operations performed on the data, the system executes these tasks in parallel rather than executing all operations at once.

Rule 3: Consider Volume of Data in Splitting

Rule 3 is the last priority, which takes into account the amount of data captured and processed when doing the splitting. As in the second rule, the maximum number of operations varies from system to system, with varying appropriate size of data for separation. Therefore, each system requires finding the appropriate size of data that can be tolerated and then providing it to the helper project algorithm which splits the huge amount of data into smaller amounts of data that are able to be processed. In the end, the helper project algorithm produces several tasks which have one category, maximum number of operations, and the proper amount of data that the system runs in parallel.

3.2 *Modifying the Current System*

For the system that is not compatible with distributed computing, a modification to the system is required to be working in the deployment environment. Distributed computing is a model in which portions of a software system are shared among multiple computers to improve both efficiency and performance [12]. This methodology suggests two versions of the system. The first one is the base version which the system starts from and is responsible for dividing the huge amount of data by executing the helper project algorithm running on top of it. Moreover, it determines when to scale up the system based on the number of jobs the helper project algorithm produces by controlling virtual servers or machines. This version deploys the separated tasks and scales up the system into a deployment environment to work in parallel. The user interface sits on this version.

The second version runs in several virtual machines in the deployment environment, so it needs to be compatible with the deployment environment. Moreover,

it has major changes that modify the current system to command a line interface rather than a user interface. This version processes the small pieces produced by the helper project algorithm in parallel, and it works in the background for the base version. Also, there are some changes in storing and retrieving the data, but that varies from system to system; it is based on the structure of the system and the dependency of tasks. This methodology suggests the data, or information, is used by all distributed jobs to be in shared storage such as cloud storage, cluster, or a shared database. In this way, the distributed system avoids any redundancy or conflict while processing. By applying this approach, few modifications in the current system are needed except the two versions: one works as the interface and the other works as the background.

3.3 Deployment Environment

After splitting the massive amount of data into small sets by the helper project algorithm, the deployment environment hosts the system in several virtual machines and processes these chunks in parallel. The International Business Machines (IBM) knowledge center defines the deployment environment as a collection of configured clusters, servers, and middleware that collaborate to provide an environment to host software modules [13]. The purpose of the deployment environment in this approach is to host and process the small tasks in parallel. There are various ways that exist to use deployment environments: it may be a network of many machines in data centers in clusters or virtual machines in cloud computing.

This approach illustrates the important aspects of the deployment environment utilized to take advantage of it and to dynamically scale up the system. Firstly, to be able to dynamically scale up the system based on the number of tasks produced by the helper project algorithm, it is necessary to control virtual servers or machines from the base version of system. The base version of the system needs the ability to start the virtual machines before the deployment in order to prepare the virtual machines for deploying and running a small task. Also, each virtual machine is responsible for shutting itself down after finishing executing the assigned tasks and producing the desired result. Secondly, in order to begin publishing the virtual machines' version onto virtual machines in the deployment environment, it places the version of the virtual machine in an accessible place for the deployment environment. Deployment environments provide service to deploying the code, scripts, or execution file, so the base version system needs to use it to control the deployment and be eligible to initiate and verify publishing in the deployment environment. Thirdly, if the system holds some data used as input, or produces data which is input for distributed tasks, then it demands to employ shared storage or a database in the deployment environment. Also, the helper project algorithm produces a reasonable size of tasks that depend on each other, so a need occurs to store all desired data as input in a shared place for accessibility to all virtual machines. Many deployment environments supply shared places to store various

types of data such as databases, blocks, or cloud storage. Hence, the virtual machine version should be able to access shared storage to read from and write to it. Thus, it grants a solution of the redundancy and conflict in processing between dependent tasks.

3.4 Database Transformation

The traditional relational database management systems face a challenge in the performance while processing large volumes of data. It is ideal to use non-relational databases, such as NoSQL, MongoDB, and Hadoop, as a solution to handle large datasets. However, it would be expensive to reimplement an existing system using a different database management system. Targeting to the systems using traditional relational databases, our approach provides the solutions to overcome the performance challenge without requesting of changing the type of databases.

The `centralDB`, located in the deployment environment, is the central database that stores the complete record of the system. It is accessible for all virtual machines to query and update as needed. After deployment, the system is hosted in several virtual machines that work in parallel. Each virtual machine contains its own database, a distributed database `localDB`, to store the results of a small task processed. The virtual machine stores its results to the `localDB` after it has accomplished processing a small job and transits the records in the `localDB` to the `centralDB`. The virtual machines then delete all records related to a processed job after moving them to the `centralDB` in order to prepare the `localDB` for the next unprocessed job. This solves the bottleneck issues caused when processing huge amounts of data.

Figure 1 illustrates the dynamic scaling methodology steps.

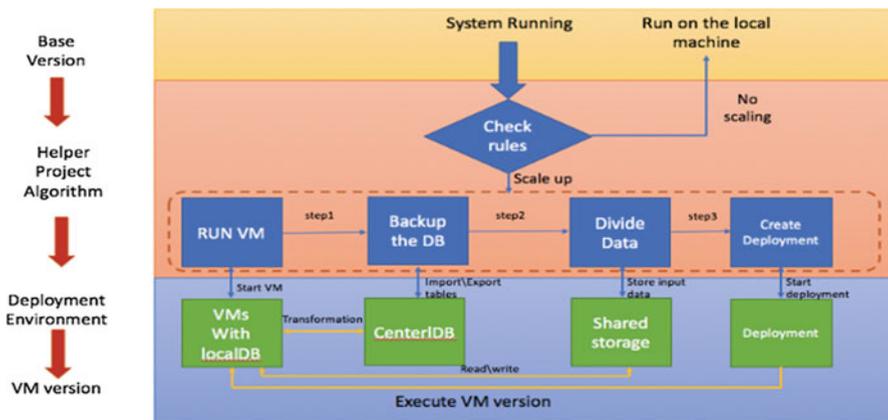


Fig. 1 Steps in dynamic scaling methodology

4 Case Study

The EASTWeb system is set up as a stand-alone application on a single computer. When initiating the system, no complications are encountered with the performance of the system; but while in progress, the data rapidly grows which raises the failure performance issue. For example, IMERG_Project uses the IMERG dataset [14] and is used as input for EASTWeb in this case study. IMERG_Project processes 3 years of data and requires 733.468 GB, and EASTWeb suffers from the poor performance – the long execution time – when processing this amount of data in a single computer. We applied our dynamic scaling methodology to the EASTWeb system to overcome performance problem.

4.1 *The Deployment Environment*

In this case study, AWS is utilized to scale up and run EASTWeb in several servers in the cloud.

AWS offers SDKs for several languages such as Java, C++, Python, Ruby, and PHP. SDKs facilitate building applications to work with Amazon services. We created four classes to make EASTWeb cooperate with Amazon S3, Amazon EC2, CodeDeploy, and RDB, and AWS SDK for Java is used to achieve scalability. The four classes are as follows: (1) `CreateInstance` class that connects to EC2 instances and controls them remotely; (2) `S3` class that controls the write, retrieve, and delete files from a bucket; (3) `CodeDeploy` class that manages and prepares the deployment of EASTWeb onto EC2 instances; and (4) `RDS` class that controls connection, starting/stopping the database in the cloud (`centralDB`), and exporting/importing data to the `centralDB`.

4.2 *Applying the Dynamic Scaling Methodology*

Three steps are involved in applying the dynamic scaling methodology to the EASTWeb system.

Applying Helper Project Algorithm

Developing the helper project algorithm to work on top of the EASTWeb system requires defining the rules and their priorities in the case of EASTWeb. The helper project algorithm is invoked after EASTWeb runs and takes the project file as input.

Applying Rule 1

In the EASTWeb system, the number of plugins is the highest priority for splitting the user-selected project into several subprojects. `CheckNumberOfPlugins()` is responsible for checking the number of plugins and making the separation based on it. If the number of plugins is more than one, then it divides the project file into several subproject files, each with one plugin. Partial code of `CheckNumberOfPlugins()` is shown in Fig. 2.

Applying Rule 2

After the helper project algorithm checks Rule 1 and finds no separation performed, it examines the number of indices as the second priority of rules. We set a maximum of five indices as the maximum number of operations. If a project has five indices or less, there is no need for splitting, and it moves to the next rule. However, it requires dividing a project if the project has more than five indices.

```

public void CheckNumOfPlugins(){
    boolean flag;
    int index;
    int partNO = 0;
    try {
        if (NumOfPlugins > 1) {
            //divide based on the number of plugins
            for (int i = 0; i < NumOfPlugins; i++) {
                //divide based on the number of years
                for(int s=0; s < startDates.size(); s++){
                    noOfIndices = pluginsInfo.get(i).GetIndices().size();
                    DivideIndices = true;
                    flag = false;
                    index = 0;
                    //divide based on the number of indices
                    while(DivideIndices){
                        // split the xml file project into several xml subproject files
                    }
                }
            }
        }
        else if (NumOfPlugins == 1) {
            CheckNumOfIndices(); //jump to rule 2
        }
    } catch (ParserConfigurationException e) {
        ErrorLog.add(Config.getInstance(),
            "problem with creating new project.", e);
    }
}

```

Fig. 2 Partial code of `CheckNumberOfPlugins()`

`CheckNumberOfIndices()` is invoked to determine the number of indices and apply the splitting. The partial code of the function is shown in Fig. 3.

Applying Rule 3

While applying Rule 3, we set the number of years of data as the size of data. The helper project algorithm checks Rule 3 at the end, and it determines the maximum volume of data in a year of data for each subproject. `CheckNumberOfYear()` function is invoked to produce various subproject files with a year of data. If a project has less than a year of data, it is not necessary to scale up the system or divide the project (Fig. 4).

Summarized in Table 1, the three functions work together to scatter the selected project and produce several subprojects.

```

public void CheckNumOfIndices() {
    boolean flag1;
    int index;
    int partNO = 0;
    try {
        if (NumOfIndices > 5) {
            noOfIndices = pluginsInfo.get(0).GetIndices().size();

            //divide based on the number of years
            for(int s=0; s < startDates.size(); s++){
                noOfIndices = pluginsInfo.get(0).GetIndices().size();
                DivideIndices = true;
                flag1 = false;
                index = 0;

                //divide based on the number of indices
                while(DivideIndices){
                    // split the xml file project into several xml subproject files
                }
            }
        }
        else if (NumOfIndices <= 5) {
            CheckNumOfYears(); //jump to rule 3
        }
    }
    catch (ParserConfigurationException e) {
        ErrorLog.add(Config.getInstance(),
            "problem with creating new project.", e);
    }
}

```

Fig. 3 Partial code of `CheckNumberOfIndices()`

```

public void CheckNumOfYears() {
    if (startDates.size() > 1) {
        int partNO = 0;
        //divide based on the number of years
        for(int s=0; s < startDates.size(); s++){
            try {
                // split the xml file project into several xml subproject files
            }
            catch (ParserConfigurationException e) {
                ErrorLog.add(Config.getInstance(),
                    "problem with creating new project.", e);
            }
        }
    }
    else{
        System.out.println("No need to divide project");
        return;
    }
}
}
}

```

Fig. 4 Partial code of CheckNumberOfYear()

Table 1 Functions of helper project algorithm

Priority	Functions	Description
1	CheckNumberOfPlugIns()	Check number of plugin >1 and scatter the input file based on the number of plugins, indices, and years. Otherwise, jump to Rule 2
2	CheckNumberOfIndices()	Check number of indices >5 and scatter the input file based on the number of indices and years. Otherwise, jump to Rule 3
3	CheckNumberOfYear()	Check number of years >1 and scatter the input file based on the number of years. Otherwise, return without splitting

Modifying the System

To run EASTWeb in the cloud environment and execute the helper project algorithm on top of it, some modifications are applied to the current system. Two versions of the system, a base version and a virtual machine (VM) version, are configured in this step, and two versions work together to scale up the system based on helper project algorithm rules. The base version runs as an interface of the system, and the VM version works in the background of the base version. The workflow of the two versions of EASTWeb is shown in Fig. 5.

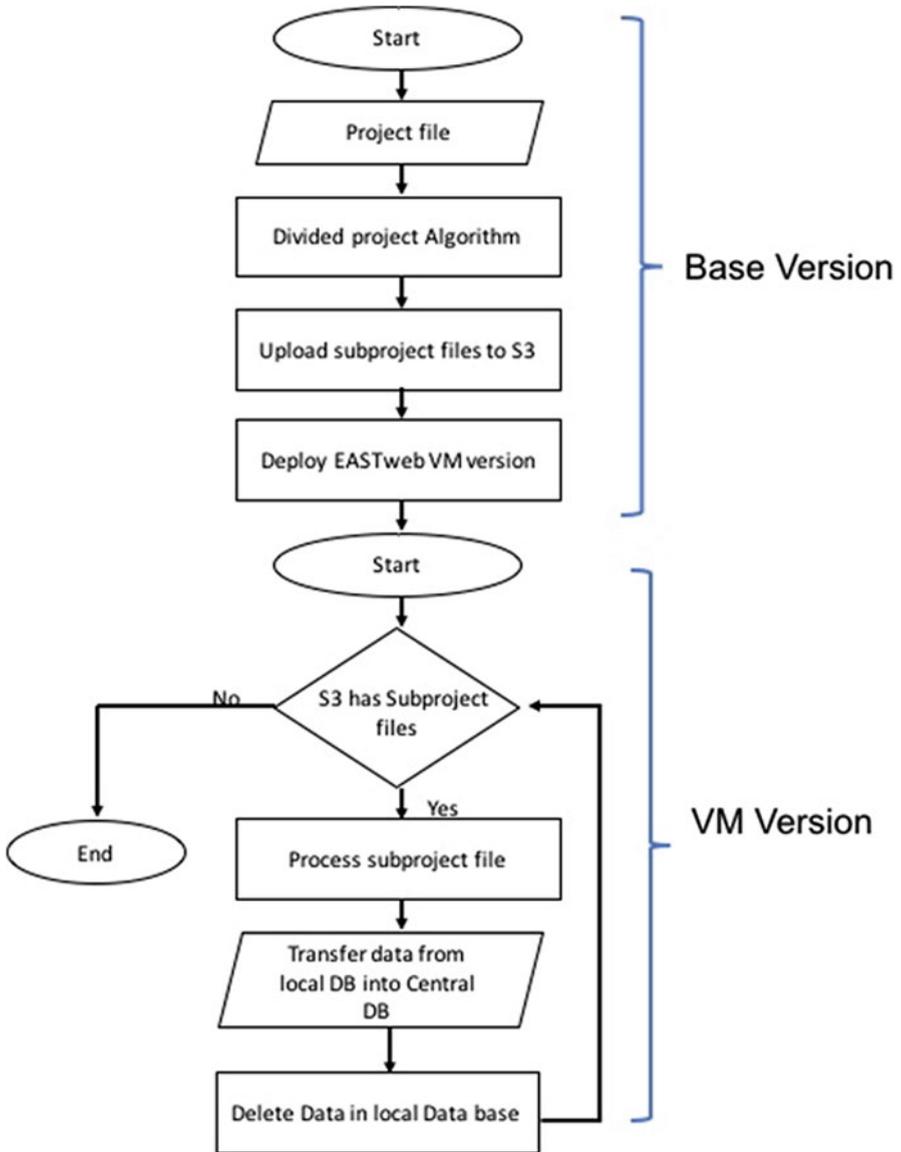


Fig. 5 The workflow of the base version and VM version of EASTWeb

Base Version

The base version of the EASTWeb connects to AWS cloud and utilizes services such as EC2, S3, and CodeDeploy. The base version reads the project file and splits the user-selected project by running the helper project algorithm, which

produces several subproject files. These files are uploaded to the cloud storage, and S3 is accessible for all VM instances by calling `WriteToS3()` function, which creates a folder called `subproject` on the S3 bucket to store subproject files. In the end, it deploys the VM version to VM instances by executing the `CreateDeployment()` function for assigning the deployment group, creating an application name for deployment, and setting up the revision location.

In the case of scaling up the EASTWeb, the resources in the cloud should be prepared for deployment and run the VM version of EASTWeb on several VM instances in the background.

Virtual Machine Version

The virtual machine version is extracted from the current system and runs on several VM instances on the cloud. Several changes have been made to be compatible with the cloud environment.

This version starts executing on a virtual machine instance after the base version creates the deployment. It takes a subproject file, produced by the helper project algorithm, as input from the cloud storage S3. As a subproject file is downloaded into a virtual machine instance, it is deleted from the cloud storage to avoid conflict between virtual machine instances. Each virtual machine instance in the cloud is able to download a subproject file and process it through EASTWeb steps. When the VM version is terminating processing the subproject files and storing its related results in the database, it rechecks the cloud storage for any unprocessed subproject files. If an unprocessed file is located, it does the same steps again. Otherwise, the VM instances shut down automatically.

The major processing steps in EASTWeb generate several intermediate files stored in the local drive. In order to avoid the data redundancy in processing steps, each virtual machine uploads the intermediate files that processing steps are produced to cloud storage.

Transforming the Database

EASTWeb utilizes the PostgreSQL database to store and manipulate the outcomes of processing data [1]. EASTWeb starts facing delays in database operations after processing 3 years of IMERGE data. Amazon Relational Database Service (Amazon RDS) provides a database over the cloud with several database engines such as PostgreSQL, Oracle, MySQL, and others. By using this service, the PostgreSQL database is created to be a central database (`centralDB`) for all virtual machine instances to store and retrieve data.

Each virtual instance has its own database as a local database (`localDB`) that works independently along with EASTWeb VM version to store and manipulate the outcomes of processing subproject data. When the EASTWeb VM version terminates all processing steps and stores the data to the `localDB`, it starts moving

the `localDB` into the `centralDB` by several steps. These steps work with an assumption of conflict of the primary keys between tables while in transformation.

Step 1: Export the six tables with conflict in primary keys from the master database related to the EASTWeb base version to make a backup of the last update of these tables and then import these tables to the `centralDB` to be up to date. This step is done before creating the deployment by the EASTWeb base version. It is essential to make the primary key of these tables concurrent.

Step 2: In the VM version, the global schema in the `centralDB` is shared among all virtual machines. While each VM processes a subproject, the global schema is updated with new records. This is important to avoid the conflict of primary keys and foreign keys.

Step 3: After a subproject is processed and stores its result in a subproject schema in the `localDB`, the subproject schema is exported to the `centralDB`. After exporting each subproject schema, it evacuates the `localDB`. This is important to prepare the `localDB` for unprocessed subprojects and avoid delays in database operations.

Step 4: Copy the six tables of the global schema from the `centralDB` after complete processing of all subproject files. Then export all subproject schemas to the master database. This step is done by the EASTWeb base version. This step updates the master database for further processing.

This approach tackles the issue of handling huge amounts of data in traditional databases. Also, it provides a solution for overlapping the primary keys of relational tables by performing these steps.

The overview of the structure of EASTWeb after the dynamic scaling methodology is applied (Fig. 6).

4.3 Results

We compare the performance of EASTWeb with and without our methodology. Each scenario runs a project called the `IMERG_Project`. This project has several entries such as a plugin, an index, and 3 years of data. The first scenario is the dynamic scaling methodology is run on the top of EASTWeb. Thus, the helper project algorithm divides this project into three subprojects based on its rules. Each subproject runs on a virtual machine to work in parallel. The second scenario is EASTWeb runs without our methodology, so the `IMERG_Project` is not separated into several subprojects. The results of the performance for each scenario are shown in Table 2.

The dynamic scaling methodology is able to decrease the running time to 10 hours as shown in scenario 1. Since our methodology divides and runs the `IMERG_Project` on several virtual machines working in parallel, the performance of EASTWeb is improved significantly.

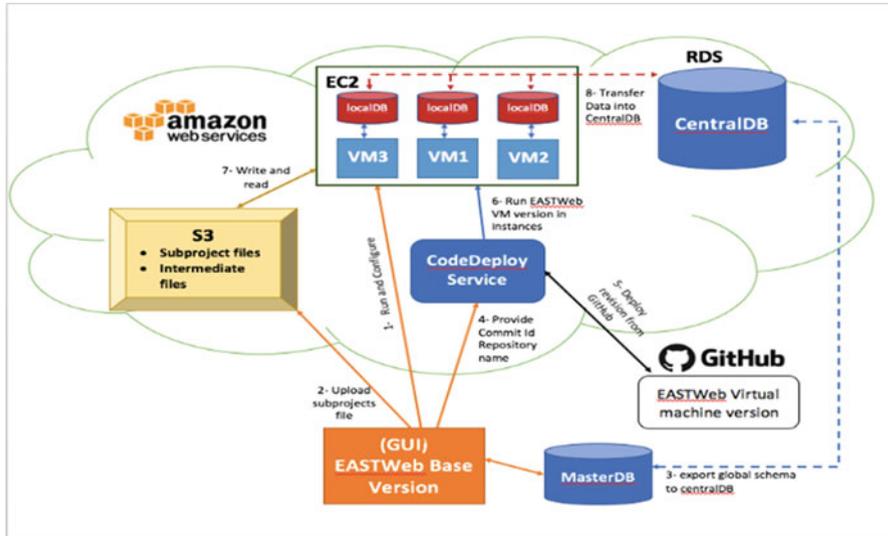


Fig. 6 Structure of modified EASTWeb

Table 2 Comparison of running IMERG_Project with and without dynamic scaling methodology

Scenarios	Scenario 1: EASTWeb with the dynamic scaling methodology	Scenario 2: EASTWeb without the dynamic scaling methodology
Project	IMERG_Project	IMERG_Project
Description	Three subprojects run on several virtual machines in parallel	A project run on the single commuter
Performance	Takes 13 hours to process 3 years of data	Takes 23 hours and 30 minutes to process 3 years of data

5 Discussion

Some research work has been done on dynamic scaling methodology. Chieu et al. [15] proposed a methodology of dynamic scaling for web applications by employing scaling indicators to decrease the running time. The scaling indicators of the web applications are number of concurrent users, number of active connections, average response times per request, and other indicators. Another dynamic scaling system [16] is focused on scaling up the system based on two parameters, user requests and response time, to enhance the quality of services of the web application. Comparing these two methodologies, our approach uses three rules as indicators for dynamic scaling. The rules are not target to specific web applications but general enough to be applied to any type of software.

The research [17] presented a model to handle analyzing a large amount of data on a cloud environment and matched the requirements of safety, easily scalable, and high efficiency. The model uses a Hadoop tool to schedule jobs and MapReduce to distribute tasks through cluster and used virtual machines to host the application services and database. Our methodology does not require a specific environment setting for deployment. Although we used specific cloud computing environment in the case study, the methodology itself supports a much broader set of deployment environment.

6 Conclusion

A novel dynamic scaling methodology is developed in this research to deal with the performance failure of systems that process huge amounts of data. This methodology involves a novel algorithm called the helper project to divide a task into several smaller tasks based on various aspects in a big data perspective. These aspects known as algorithm rules are prioritized from high to low for dividing the project. Also, the helper project decides whether it is necessary to scale up the system based on the number of tasks produced after separation. The methodology applies the necessary scaling up of the system to run in several virtual machines in the deployment environment and addresses the bottlenecks of relational databases during processing huge amounts of data by suggesting a database transformation approach.

The methodology is applied to the EASTWeb system, which suffers a severe performance issue (extreme long responding time) in processing huge amount of data. We compared the speed of running the same project in EASTWeb with and without dynamic scaling methodology, and results show that applying the dynamic scaling methodology significantly improved the performance.

As a direction of future work, the dynamic scaling methodology will work along with other deployment environment to be more flexible and general.

There are several deployment environments such as cluster and virtualization [18]. Therefore, utilizing the dynamic scaling methodology in other deployment environments will create a more general and flexible methodology. Using virtualization in VMware [19] along with the dynamic scaling methodology will be the next step to enhance our work.

Our methodology focuses on horizontal scaling to improve the performance of big data systems. Horizontal scaling capability increases the resources, such as hardware or software, to improve performance [20]. The helper project is able to increase and decrease the virtual machine instances based on the number of tasks to be processed by the system. Thus, to improve the dynamic scaling methodology, an algorithm will be developed to manage the scaling vertically. The vertical scaling ability increases resources such as increasing memory space and CPU to a machine [20]. To apply the vertical scaling to our methodology, a need exists to identify the proper resources for each task. Applying scaling in both directions, horizontally and vertically, will provide better enhancement in the performance of big data systems.

Acknowledgments This work was supported by the National Institute of Allergy and Infectious Diseases (Grant Number R01AI079411). Development of the EASTWeb software was supported by NASA through the Advancing Collaborative Connections for Earth System Science Program (Grant Number NNX14AI37A).

References

1. L. Yang, L. Guo, Y. Guo, An efficient and performance-aware big data storage system, in *International Conference on Cloud Computing and Services Science*, (Springer, Cham, 2012), pp. 102–116
2. Y. Liu, J. Hu, I. Snell-Feikema, M.S. VanBemmel, A. Lamsal, M.C. Wimberly, Software to facilitate remote sensing data access for disease early warning systems. *Environmental Modeling and Software* **74**, 247–257 (2015)
3. The epidemiological applications of spatial technologies project website, <https://eastweb.sdstate.edu/>
4. I.A.T. Hashem, I. Yaqoob, N.B. Anuar, S. Mokhtar, A. Gani, S.U. Khan, The rise of “big data” on cloud computing: Review and open research issues. *Inf. Syst.* **47**, 98–115 (2015)
5. S. Ostermann, A. Iosup, N. Yigitbasi, R. Prodan, T. Fahringer, D. Epema, A performance analysis of EC2 cloud computing services for scientific computing, in *In International Conference on Cloud Computing*, (Springer, Berlin, Heidelberg, 2009), pp. 115–131
6. Microsoft Azure, <https://azure.microsoft.com/en-us/>. Last accessed on 15 June 2020
7. Google Cloud Platform, <https://cloud.google.com>. Last accessed on 15 June 2020
8. Amazon Web Service, <https://aws.amazon.com>. Last accessed on 15 June 2020
9. What is AWS EC2? <https://www.sumologic.com/aws/what-is-aws-ec2/>. Last accessed on 15 June 2020
10. AWS CodeDeploy (Amazon Web Services CodeDeploy), <http://searchitoperations.techtarget.com/definition/AWS-CodeDeploy-Amazon-Web-Services-CodeDeploy>. Last accessed on 15 June 2020
11. P. Bihani, S.T. Patil, A comparative study of data analysis techniques. *International Journal of Emerging Trends & Technology in Computer Science* **3**(2), 95–101 (2014)
12. Distributed computing, <http://whatis.techtarget.com/definition/distributed-computing>. Last accessed on 15 June 2020
13. Deployment environments, https://www.ibm.com/support/knowledgecenter/en/SSTLXK_7.5.1/com.ibm.wbpm.ref.doc/help_nd/index.html. Last accessed on 15 June 2020
14. Global precipitation measurement IMERG, <https://gpm.nasa.gov/category/keywords/imerg>. Last accessed on 15 June 2020
15. T.C. Chieu, A. Mohindra, A.A. Karve, A. Segal, Dynamic scaling of web applications in a virtualized cloud computing environment. in *E-Business Engineering, 2009. ICEBE'09. IEEE International Conference on, IEEE, 2009*, pp. 281–286
16. S. Pandey, W. Voorsluys, S. Niu, A. Khandoker, R. Buyya, An autonomic cloud environment for hosting ECG data analysis services. *Futur. Gener. Comput. Syst.* **28**(1), 147–154 (2012)
17. Y. Pradhananga, S. Karande, C. Karande, High performance analytics of big data with dynamic and optimized hadoop cluster, in *Advanced Communication Control and Computing Technologies (ICACCCT), 2016 International Conference on, IEEE, 2016*, pp. 715–720
18. Deployment environments, https://www.ibm.com/support/knowledgecenter/SSFPJS_8.5.0/com.ibm.wbpm.ref.doc/help_nd/index.html. Last accessed on 15 June 2020
19. VMware, <https://www.vmware.com>. Last accessed on 15 June 2020
20. Horizontal scalability, <http://searchcio.techtarget.com/definition/horizontal-scalability>. Last accessed on 15 June 2020

Part VI
Software Engineering Research, Practice,
and Novel Applications

Technical Personality as Related to Intrinsic Personality Traits



Marwan Shaban, Craig Tidwell, Janell Robinson, and Adam J. Rocke

1 Introduction

Intrinsic personality is a set of characteristics that help identify tendencies in personal behavior. It is typically evaluated using standardized surveys to measure personality traits. Technical personality is a profile of the broad technical preferences of someone working in a technology-related field. For instance, one technical personality trait is the preference for formal or informal documentation. By measuring both intrinsic and technical personalities in a sample of IT workers, it can be determined whether a correlation exists between the two. This research attempts to identify whether personality traits influence technical preferences. There are stereotypes in IT, e.g., that tech workers are more likely to be introverted and to prefer tasks that are more isolated and independent. This research can also help confirm or dispel such stereotypes.

The hypothesis is that there is no correlation between technical personality and intrinsic personality.

H_0 : There is no correlation between Technical Personality and Intrinsic Personality.

H_A : There is correlation between Technical Personality and Intrinsic Personality.

M. Shaban (✉) · C. Tidwell · J. Robinson · A. J. Rocke

Seminole State College, Sanford, FL, USA

e-mail: shabanm@seminolestate.edu; tidwellc@seminolestate.edu; robinsonj@seminolestate.edu; rockea@seminolestate.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_42

597

2 Previous Research

Previous researchers have used the Big Five model of personality as it relates to the job performance, job satisfaction, career success, life satisfaction, and academic performance of IT professionals (Lounsbury, Studham, Steel, Gibson, and Drost) [11]. Rodrigues and Rebelo [14] examined the correlation between job performance of software engineers and their Big Five profiles, as well as their level of proactive personality. Lounsbury, Sundstrom, Levy, and Gibson [12] examine the Big Five traits among IT professionals as opposed to professionals at large, in a study of 73,000 individuals, and relate personality differences between the two groups. Another study about the characteristics of Information Technology professionals concentrates on burnout, turnover intentions, and strategies for retaining these highly skilled professionals (Paré and Tremblay) [13]. Eckhardt [9] studies the Big Five trait differences between the specialties of IT professionals, and whether each difference is a predictor of turnover intentions.

Current research using tests other than the Big Five and the Myers–Briggs Type Indicator (MBTI) is limited. Bishop-Clark [2] examines previous studies analyzing several personality traits as they relate to computer programming and phases thereof. Included in the personality traits examined were MBTI factors, field dependence/independence, analytic/holistic, impulsivity/reflectivity, and divergent thinking. Capretz et al. [7] performed empirical studies that correlated MBTI types' distribution among software engineers. The authors aimed to find a link between personality traits and role preferences in a software life cycle. Cruz et al. [8] provide a survey of studies on personality in software engineering. They show findings on questions such as which personality types are most common in software engineering, and what effects personality types have on effectiveness of software engineering. The study of software developers' personalities to determine performance and motivation has been ongoing. Wiesche [16] examines recent research into personality types' impact on success at different software engineering tasks. Furthermore, researchers have discussed the use of MBTI to study various human factors in technology-related areas. Capretz and Ahmed [5] advocate mapping MBTI factors to job descriptions to analyze the fit of workers to tasks within software engineering. Varona et al. [15] examine sixteen studies that analyze MBTI profiles of software engineers and derive trends over time of each personality type within the field of software engineering, while Woszczyński [17] and Bishop-Clark et al. [3] analyze MBTI profiles of programming students and correlate personality types with success at computer programming.

3 Intrinsic and Technical Personality Traits

Intrinsic personality is commonly evaluated using the Big Five personality traits. The five factors are Extroversion, Agreeableness, Conscientiousness, Emotional Stability, and Intellect/Imagination.

The four technical personality traits identified in this research are defined below. These all are dichotomies that are pervasive in technical teams, though not unique to technical teams.

1. **Formal.** Formal individuals tend to prefer comprehensive documentation over FAQs (responses to a list of frequently asked questions), predefined processes over spontaneous interactions, and written communication such as email over casual communications.
2. **Enterprising.** More enterprising tech workers tend to prefer working on new projects over sustaining or maintaining existing systems. They tend to enjoy expanding the impact of technology and working with innovative new technologies as opposed to technologies that are well established. They do not typically enjoy the routine although necessary work associated with maintaining well-established/legacy systems.
3. **Collaborative.** Collaborative tech workers tend to prefer working in groups as opposed to working on projects individually. Their creativity is prompted by brainstorming with peers, and they are less likely to enjoy doing technical work, especially work requiring creativity, by themselves. Less collaborative individuals tend to prefer working solo and may consider working in groups to be distracting. Collaborative supervisors tend to be more people-oriented and less process-oriented.
4. **Expeditious.** Tech workers who are more expeditious tend to push projects forward striving to reach goals and overcome obstacles, technical or otherwise. Less expeditious workers tend to push for quality over quantity. Here, the concepts of quality and productivity roles are introduced, quality roles being, e.g., quality analyst and system architect, and productivity roles being, e.g., software developer and manager. We investigate whether expeditiousness varies between these two role types.

4 Testing Methodology and Data Collection

A survey was developed to collect data from a diverse group of audiences in 2018 and 2019. These audiences included a technology industry advisory board, a college information technology department, and multiple junior- and senior-level college students. It was also distributed via social media.¹

¹At the time of publication, the survey could be found at <http://bit.ly/TechPersonality>.

The survey contains two parts. The first part is a standard Big Five personality profile with five traits and ten questions for each trait, for a total of 50 questions. These survey prompts and evaluation criteria were selected from the open source International Personality Item Pool (IPIP scale), based on [10]. The second part of the survey is ten questions for each of the four technical traits defined above for a total of 40 questions. The survey was presented to individuals with diverse technical backgrounds to identify their intrinsic and technical personalities. After completion of the anonymous online survey, the users were provided with information about each trait that was measured, as well as how their responses are compared to the averages. The survey instructions provided are listed below,

Describe yourself as you generally are now, not as you wish to be in the future. Describe yourself as you honestly see yourself, in relation to other people you know of the same sex as you are, and roughly your same age. Indicate for each statement whether it is 1. Very Inaccurate, 2. Moderately Inaccurate, 3. Neither Accurate nor Inaccurate, 4. Moderately Accurate, or 5. Very Accurate as a description of you.

Examples of statements in the evaluation of intrinsic personality include:

- I am the life of the party.
- I feel little concern for others.
- I am always prepared.
- I get stressed out easily.

Examples of statements in the evaluation of technical personality include:

- Documentation should be formal.
- I enjoy building new systems.
- Creating systems is best done in a group setting.
- Quality is more important than quantity.

Results are reported to the respondent upon submission of the survey. The following is a sample narrative. Here, only two traits are shown as an example:

It's important to note that there is no right or wrong personality type. Your scores simply reflect your own style and personality.

Intellect/Imagination

Your score is 0.33. The range is -1 to 1 , 1 having the highest intellect/imagination.

The average for all respondents is 0.40, with a standard deviation of 0.31. This means that 70% of respondents scored between 0.09 and 0.71.

Also described as openness to experience, this is one of the domains which are used to describe human personality in the Five Factor Model. Openness involves five facets, or dimensions, including active imagination (fantasy), aesthetic sensitivity, attentiveness to inner feelings, preference for variety, and intellectual curiosity. A great deal of psychometric research has demonstrated that these facets or qualities are significantly correlated. Read more...

Formality

Your score is 0.12. The range is -1 to 1 , 1 being the most formal.

The average for all respondents is 0.16, with a standard deviation of 0.18. This means that 70% of respondents scored between -0.02 and 0.34.

Formal tech workers tend to prefer comprehensive documentation over FAQs, predefined process over spontaneous interactions, and formal communications (e.g., email) over casual communications (e.g., messaging or face-to-face).

5 Data Analysis

The five intrinsic personality traits and four technical preferences are evaluated. Each of the nine traits produced a bell-shaped curve as shown in Fig. 1.

On a scale of -10 to 10 , the averages and standard deviations are shown in Fig. 2.

The correlation coefficients for all nine traits are shown in Fig. 3.

High significance is identified where correlations were above 0.2 in the above table. With a standard alpha value of 0.05 , the p -values are shown in Fig. 4.

The following scale was used to evaluate correlation:

- $0-0.2$: negligible
- $0.2-0.4$: modest
- $0.4-0.6$: moderate
- $0.6-0.8$: significant
- $0.8-1$: high.

The results show that no moderate, significant, or high correlations exist. However, a few modest correlations are identified:

- Expeditiousness has a modest negative correlation with agreeableness, intellect/imagination, formality, and collaborativeness. In other words, expeditious people are less likely to be open to new experiences (intellect/imagination), to be formal, or to go along with what other people want.
- Collaborativeness has a modest correlation with extroversion and agreeableness.

The following charts evaluate the data in terms of the demographics provided (Figs. 5, 6, 7, 8, 9, 10, 11, 12, 13, 14).

Several interesting facts emerge from the data above, including:

- Formality increases and expeditiousness decreases as education level increases.
- Expeditiousness does not vary between quality and productivity roles.
- Women in tech roles are typically more extraverted, more agreeable, and less enterprising than men.
- Technology workers in productivity roles tend to be more collaborative and extraverted than those in quality roles.
- Older tech workers are more conscientious and more introverted.

Clusters can be found in the technical personality data and are shown in Figs. 15 and 16.

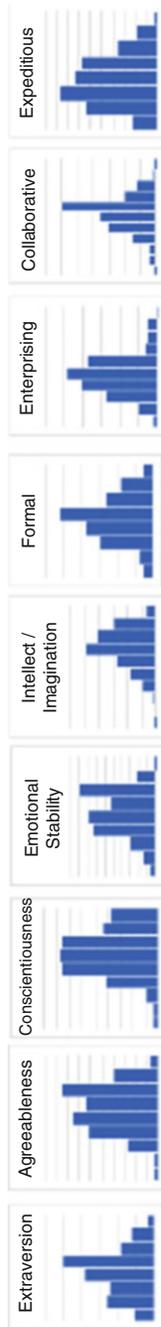


Fig. 1 Distribution of scores for each trait

	Extra-version	Agreeableness	Conscientiousness	Emotional Stability	Intellect/Imagination	Formal	Enterprising	Collaborative	Expeditious
Avg (-10 to 10)	-0.4	3.8	4.2	2.0	4.0	1.6	0.5	0.6	-3.2
Std. Deviation	4.32	3.07	2.99	3.85	3.09	1.79	1.51	2.10	2.08

Fig. 2 Average and standard deviation for each trait’s scores

Correlation	Extra-version	Agreeableness	Conscientiousness	Emotional Stability	Intellect / Imagination	Formal	Enterprising	Collaborative	Expeditious
Extraversion	1.00								
Agreeableness	0.26	1.00							
Conscientiousness	0.01	0.17	1.00						
Emotional Stability	0.20	0.16	0.18	1.00					
Intellect/Imagination	0.24	0.15	0.18	0.20	1.00				
Formal	-0.07	0.16	0.19	0.13	0.06	1.00			
Enterprising	0.01	-0.16	-0.17	0.04	0.17	-0.10	1.00		
Collaborative	0.29	0.35	0.10	0.14	-0.05	0.15	-0.18	1.00	
Expeditious	-0.02	-0.32	-0.19	-0.18	-0.25	-0.25	0.00	-0.30	1.00

Fig. 3 Correlation coefficients for each pair of traits

Significance	Extra-version	Agreeableness	Conscientiousness	Emotional Stability	Intellect / Imagination	Formal	Enterprising	Collaborative	Expeditious
Extraversion	1.00								
Agreeableness	0.00	1.00							
Conscientiousness	0.91	0.02	1.00						
Emotional Stability	0.01	0.03	0.02	1.00					
Intellect/Imagination	0.00	0.04	0.01	0.01	1.00				
Formal	0.37	0.03	0.01	0.09	0.41	1.00			
Enterprising	0.86	0.03	0.02	0.62	0.02	0.18	1.00		
Collaborative	0.00	0.00	0.19	0.06	0.47	0.04	0.02	1.00	
Expeditious	0.80	0.00	0.01	0.02	0.00	0.00	0.99	0.00	1.00

Fig. 4 p-Values for the correlations in Fig. 3

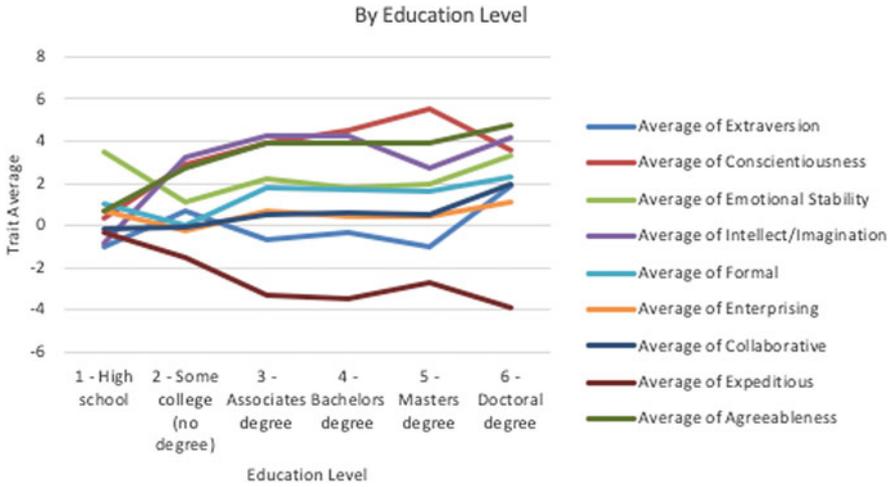


Fig. 5 Trait scores by education level

Fig. 6 Sample sizes by education level

<i>Sample Size</i>	
<i>1 - High school</i>	3
<i>2 - Some college (no degree)</i>	12
<i>3 - Associates degree</i>	77
<i>4 - Bachelors degree</i>	73
<i>5 - Masters degree</i>	16
<i>6 - Doctoral degree</i>	5
Total	186

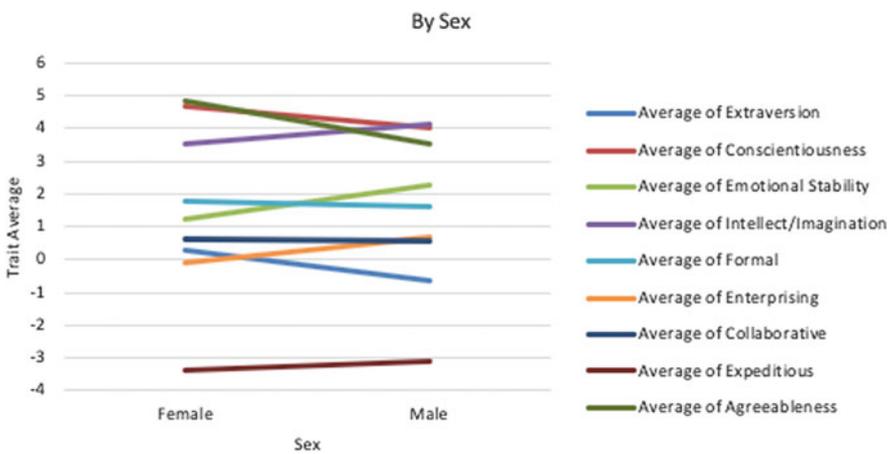


Fig. 7 Trait scores by sex

Fig. 8 Sample sizes by sex

<i>Sample Size</i>	
<i>Female</i>	44
<i>Male</i>	142
Total	186

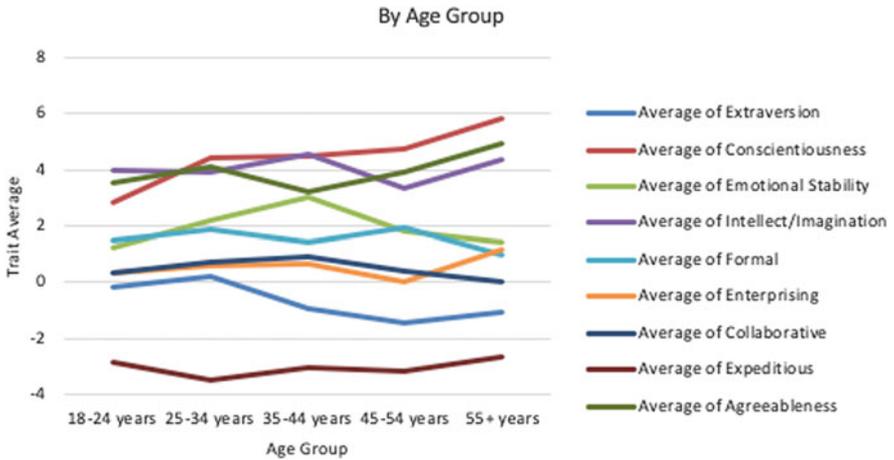


Fig. 9 Trait scores by age group

Fig. 10 Sample sizes by age group

<i>Sample Size</i>	
<i>18-24 years</i>	45
<i>25-34 years</i>	72
<i>35-44 years</i>	32
<i>45-54 years</i>	25
<i>55+ years</i>	12
Total	186

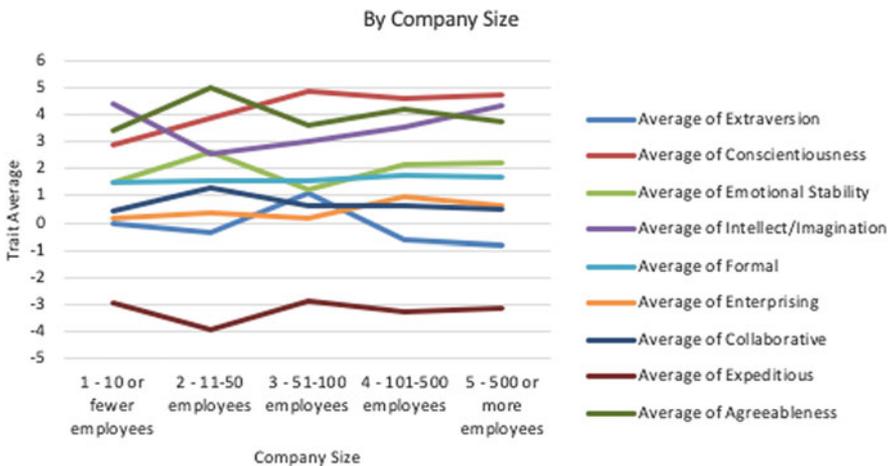


Fig. 11 Trait scores by company size

Fig. 12 Sample sizes by company size

<i>Sample Size</i>	
<i>1 - 10 or fewer employees</i>	48
<i>2 - 11-50 employees</i>	18
<i>3 - 51-100 employees</i>	13
<i>4 - 101-500 employees</i>	25
<i>5 - 500 or more employees</i>	82
Total	186

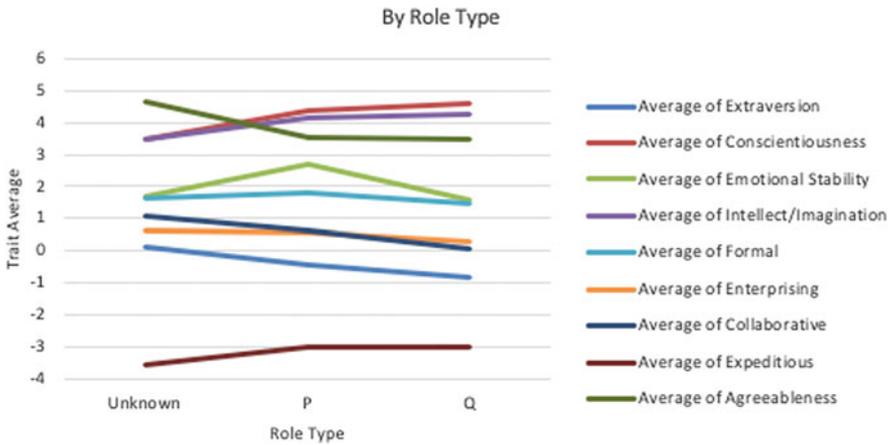


Fig. 13 Trait scores by role type

Fig. 14 Sample sizes by role type

<i>Sample Size</i>	
<i>Unknown</i>	55
<i>P (Productivity)</i>	67
<i>Q (Quality)</i>	64
Total	186

6 Conclusion

The research data presented shows that there is not a significant correlation between intrinsic and technical personalities. Modest correlations do exist between the extraverted and collaborative personality traits, between the agreeable and collaborative personality traits, and a modest negative correlation exists between the agreeable and expeditious personality traits. The data further shows that expeditiousness does not vary between quality and productivity roles.

Future work includes obtaining additional data as a larger sample size would result in higher confidence in the correlations. Also, statements for evaluating the Expeditious trait will be rewritten to de-emphasize the choice of quality vs. quantity.

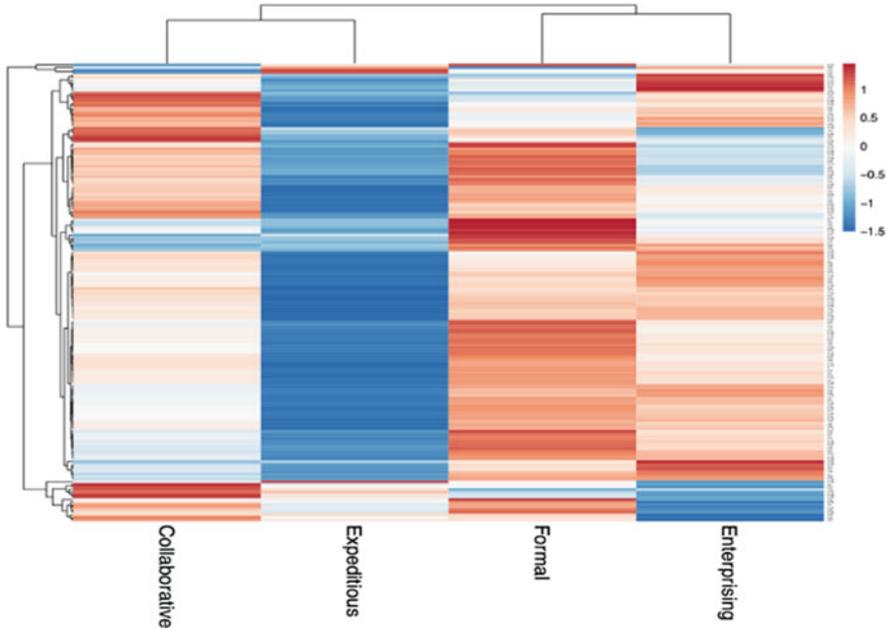


Fig. 15 Clusters in technical personality trait data

Cluster	Percent	Nickname	Collaborative	Expeditious	Formal	Enterprising
1	2%		Low	High	Medium	Medium
2	4%		Medium	Low	Medium	High
3	7%		High	Low	Medium	High
4	3%	Idealist	High	Low	Medium	Low
5	16%	Worker bee	High	Low	High	Medium or Low
6	6%	Gatekeeper	Low	Low	High	Medium
7	16%	Thinker	Medium	Low	Medium	High
8	16%	Soldier	Medium	Low	High	Medium
9	20%		Medium	Low	High	High
10	10%	Team player	High	Medium	Any	Low

Fig. 16 Clusters in technical personality trait data

This may provide a more nuanced understanding of technical personality, perhaps with multiple facets, and may also reveal multiple underlying factors.

References

1. K. Behrenbruch, M. Söllner, J. Leimeister, L. Schmidt, Understanding diversity—the impact of personality on technology acceptance, in *IFIP Conference on Human-Computer Interaction*, September (Springer, Berlin, Heidelberg, 2013), pp. 306–313
2. C. Bishop-Clark, Cognitive style, personality, and computer programming. *Comput. Hum. Behav.* **11**, 241–260 (1995)
3. C. Bishop-Clark, D. Wheeler, The Myers-Briggs personality type and its relationship to computer programming. *J. Res. Comput. Educ.* **26**, 358–370 (1994)

4. L. Capretz, Personality types in software engineering. *Int. J. Hum.-Comput. Stud.* **58**, 207–214 (2003)
5. L. Capretz, F. Ahmed, Making sense of software development and personality types. *IT Professional* **12**(1), 6–13 (2010)
6. L. Capretz, F. Ahmed, Why do we need personality diversity in software engineering? *SIGSOFT Softw. Eng. Notes* **35**, 1–11 (2010)
7. L. Capretz, D. Varona, A. Raza, Influence of personality types in software tasks choices. *Comput. Hum. Behav.* **52**, 373–378 (2015)
8. S. Cruz, F.Q. da Silva, L.F. Capretz, Forty years of research on personality in software engineering: a mapping study. *Comput. Hum. Behav.* **46**, 94–113 (2015)
9. A. Eckhardt, S. Laumer, C. Maier, T. Weitzel, The effect of personality on it personnel's job-related attitudes: establishing a dispositional model of turnover intention across it job types. *J. Inf. Technol.* **31**(1), 48–66 (2016)
10. L.R. Goldberg, The development of markers for the big-five factor structure. *Psychol. Assess.* **4**(1), 26–42 (1992)
11. J. Lounsbury, R. Studham, R. Steel, L. Gibson, A. Drost, Personality traits and career satisfaction of information technology professionals, in *Handbook of Research on Contemporary Theoretical Models in Information Systems* (Swansea University, Wales, 2009), pp. 529–543
12. J. Lounsbury, E. Sundstrom, J. Levy, L. Gibson, Distinctive personality traits of information technology professionals. *Comput. Inf. Sci.* **7**(3), 38 (2014)
13. G. Par, M. Tremblay, The influence of high-involvement human resources practices, procedural justice, organizational commitment, and citizenship behaviors on information technology professionals' turnover intentions. *Group Org. Manage.* **32**(3), 326–357 (2007)
14. N. Rodrigues, T. Rebelo, Incremental validity of proactive personality over the big five for predicting job performance of software engineers in an innovative context. *Revista de Psicología del Trabajo y de las Organizaciones* **29**(1), 21–27 (2013)
15. D. Varona, L.F. Capretz, Y. Piñero, A. Raza, Evolution of software engineers personality profile. *SIGSOFT Softw. Eng. Notes* **37**(1), 1–5 (2012)
16. M. Wiesche, H. Kremer, The relationship of personality models and development tasks in software engineering, in *Proceedings of the 52nd ACM Conference on Computers and People Research*, New York, NY, 2014, SIGSIM-CPR 14 (Association for Computing Machinery, New York, 2014), pp. 149–161
17. A.B. Woszczyński, T.C. Guthrie, S. Shade, Personality and programming. *J. Inf. Syst. Educ.* **16**(3), 293–299 (2005)

Melody-Based Pitch Correction Model for a Voice-Driven Musical Instrument



John Carelli

1 Introduction

The motivation for this work sprang from earlier research involving the development of a voice-driven musical instrument [1]. In that work, a musical instrument was designed that can recognize a stream of audio pitches a singer is producing and then uses the information to drive an independent musical instrument. The goal was to control, vocally, a separate, recognizable, virtual instrument, such as a trumpet or trombone, and faithfully reproduce the notes that the singer is singing in that other instrument. Further, this was to be done in real time so that it could be used in a live musical performance. The driven instrument could either be a built-in, sampled instrument or a virtual instrument hosted in a separate application driven via the MIDI communication protocol [2].

By design, only pitch information was derived from the voice. To allow additional interpretive control of the musical performance, including such features as volume, instrument selection, and expression adjustments in the target instrument itself, a separate physical controller was designed which could be manipulated during the performance. Ultimately, the singer ends up “playing” the instrument with both their voice and the physical controller.

There are a number of challenges associated with such an effort, especially since the primary goal was to make the device usable in real time. Among these are the detection and stabilization of the singer’s pitch and the inherent latency in that process. Of particular interest are the challenges associated with extracting the correct musical note, i.e., the note the singer intends, from the pitch stream

J. Carelli (✉)

Computer Science and Information Technology, Kutztown University of Pennsylvania, Kutztown, PA, USA

e-mail: carelli@kutztown.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_43

609

that is being produced. Inaccuracies in the singer's pitch, both during attack and sustain, as well as vibrato and latency in the pitch extraction process all contribute to complications in correctly interpreting and producing the correct note in the target instrument [3, 4]. Again, this must be accomplished in real time, so that the target instrument does not "fall behind" in the performance. These issues were discussed, together with approaches for managing them, in the prior work. Here will be examined possible techniques for improving on correct note recognition.

2 Overview

The main focus of this effort is to examine potential mechanisms for overcoming inaccurate musical note production by the singer. Beyond the latency and pitch stability issues dealt with previously is the question of what can be done after the pitch stream has been stabilized, but the singer's pitch is between notes, that is, a bit sharp or flat. The process of addressing this is often referred to as "auto-tuning" and is commonly used to correct a singer's pitch in music recording studios. Indeed, sophisticated tools exist for that very purpose [5]. Of course, in that scenario, the "pitch correction" is done after the recording has been made, that is, not in real time.

It is also possible to attempt auto-tuning in live performance, which is more relevant to this situation, and devices and software exist for that purpose as well [6]. Often, they are given information about the musical key, or about which notes to prohibit. In addition, the user is sometimes given some programmatic control over the extent to which the auto-tuner can modify the output as it is entirely possible to "correct" to the wrong note. The goal, of course, in such situations, whether live or pre-recorded, is to modify the singer's actual waveform to make it sound more pleasing or "in tune."

For the purpose of this work, the author is envisioning a performance situation wherein the instrument is not given any information about musical key or note usage. The only available information is the melody line that the singer is producing. This avoids the need for the singer, or anyone else, to adjust the instrument by entering a key or other programming information during a performance, which can be a limiting factor in using existing approaches to live auto-tuning. Note also that there is no need in this application to produce an output vocal waveform, as the voice is not the end product, so any additional latency that would be incurred by using an existing auto-tuning technique to tune the voice itself before pitch detection is attempted can be avoided.

The goal, instead, is to use a short window of the most recently sung notes to infer the likely keys, or tonal center. Given that, if the singer's next stable pitch is "between notes," the following questions are relevant. Which of those two notes is more likely? Can this be used reliably to select the "correct" note? Can this be done with a small enough window of recent notes to make it sufficiently responsive – and,

finally, can this be accomplished in real time for use in a live performance? These are the goals and questions to be addressed in this work.

3 Approach

As stated, the goal in this effort is to decide, in real time, which of the two neighboring musical notes is more likely to be the correct note in situations where a singer's pitch is between notes, that is, sharp or flat. In order to do this, a small window of recently sung notes will be used to infer a tonal center. For the purpose of this discussion, a tonal center is taken to be a key or set of, presumably related, keys that the melody is likely to be in. Within a given tonal center, certain notes will be expected to be more likely than others. This information will be used to drive decisions about when, and how much, to auto-correct the final output pitch.

To accomplish this, a mechanism for inferring expected note usage, or note probability, needs to be developed, and there is, indeed, existing research into the distribution of note usage in major and minor keys. Note distribution, or note weighting, in this context, is taken to mean the relative frequency of occurrence of each of the 12 possible musical tones. It can be derived from a song by simply computing the sum of the durations that occurs in the song for each of those notes. Conceptually, this can include both notes in the melody as well as chord/accompaniment notes. The result is a listing of the durations for each individual note. It may or may not be normalized. Each note's relative value is then reflective of the likelihood of that note's occurrence.

Some of the first work in this area is a key-finding model, by Krumhansl and Schmuckler, that compared the actual distribution of notes in a given song to idealized distributions [7]. Two such 12-tone idealized distribution arrays were products of the research, one representing major keys and the other minor keys. The two distributions are not key-specific. The first array entry in either distribution is simply the tonic note in a given key. Distribution values, both as published and normalized, are displayed in Table 1. Works by other researchers have generated similar distributions using various analysis techniques.

To determine a song's key, these idealized distributions are correlated against the actual song's note distribution for each of the 12 possible major or minor keys (24 total) by simply transposing distributions to the appropriate tonic note for each possible key. This can be done with a Pearson correlation, also known as product-moment correlation coefficient, which produces a correlation value between -1 and 1 , with a value of 1 being absolute correlation and -1 , anticorrelation. Mathematically, the correlation value, r_{xy} , for the distributions x and y is given by the following eq. [8]:

$$r_{xy} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \quad (1)$$

Table 1 Note weighting in major and minor keys

Note number	Krumhansl/Schmuckler		Key model	
	Major	Major	Major	Minor
1	6.35 (0.152)	6.33 (0.142)	0.2689	0.2521
2	2.33 (0.056)	2.68 (0.060)	0.0229	0.0245
3	3.48 (0.083)	3.52 (0.079)	0.1153	0.1179
4	2.33 (0.056)	5.38 (0.121)	0.0179	0.1504
5	4.38 (0.105)	2.60 (0.058)	0.1187	0.0117
6	4.09 (0.098)	3.53 (0.079)	0.1037	0.0877
7	2.52 (0.060)	2.54 (0.057)	0.0135	0.0108
8	5.19 (0.124)	4.75 (0.107)	0.1184	0.1455
9	2.39 (0.057)	3.98 (0.089)	0.0256	0.0825
10	3.66 (0.087)	2.69 (0.060)	0.0951	0.0171
11	2.29 (0.055)	3.34 (0.075)	0.0451	0.0744
12	2.88 (0.069)	3.17 (0.071)	0.0552	0.0254

Values in parenthesis are normalized

where n is the sample size, 12 in this case and x_i and y_i are the individual note weights from the song and idealized distributions (major or minor), respectively. \bar{x} and \bar{y} are the arithmetic means of the two distributions. The indices i identify specific notes in the 12-tone scale. So, for example, to correlate the song notes against C-major, “C” would be note number 1 from the table, “C#” would be 2, and so forth. The weights in the major column would be used in the above formula together with the song note weights to compute r_{xy} for the given key, C-major. This is done for all 24 major and minor keys, with appropriate transpositions. The major or minor key with the highest correlation value is deemed to be the song key.

The approach taken here will be to create a model which will take, as its input, a note distribution extracted from a sampling of recently sung notes, infer a tonal center – a set of keys – in a manner similar to the Krumhansl/Schmuckler approach, and produce a distribution of expected note usage for that tonal center. This, as described above, will inform pitch correction decisions. A byproduct of this model development is a new set of idealized major and minor distributions created specifically for analysis of melody alone.

A method for generating the suggested model and, perhaps more importantly, a mechanism for testing the accuracy of next-note predictions needed to be developed. The approach taken was to make use of a large library of songs from which to extract data for both model development and testing. Specifically, a collection of songs in MusicXML format was employed. The collection is available from a website called MuseScore [9] (it was originally created at a, now-defunct, website called Wikifonia). It contains over 6000 songs, supplied by a user community mainly as lead sheets, and thus includes musical key, melody, and chord change information for each song – the first two being what are needed for this analysis.

4 Model Development

4.1 Data Preparation

Prior to model development, an analysis was conducted of the songs in the library to ensure data integrity. This was deemed necessary because, as mentioned, the songs were supplied by users and entries were found to vary in quality and even in overall correctness. Both this effort and the subsequent analyses to be described below were aided substantially by the use of the powerful Python-based *music21* toolkit developed at MIT for the specific purpose of performing computer-based analysis of music [10]. In particular, *music21* has the capability of not only reading song files in a variety of formats, including MusicXML, but supporting musical key determination through the use of the basic technique just described. The Krumhansl/Schmuckler distribution is built-in, as are distributions from several other researchers.

Using these capabilities, the key for each song was extracted using five such distributions contained in the toolkit [7, 11–14]. These were all compared to the stated key in the song. Only songs for which all analyses agreed, both with each other and with the listed song key, were retained (about 70% of the total). In cases where a song was in multiple keys, i.e., transpositions occur, each transposed section was analyzed separately. So, further references to songs in the remainder of this document should, more accurately, be interpreted as song sections in a given key. The goal was to separate songs in major and minor keys for the extraction of note weightings, or distributions, in the next step of the process.

It should also be pointed out that most of the songs in the library were in major keys, approximately 85% of them. In order to provide balance between the numbers of songs in major and minor keys, additional songs in minor keys were generated algorithmically from songs found to be strongly in a major key. This was accomplished by adjusting the appropriate melody notes to put the song in the related minor key (e.g., transpose from C-major to C-minor). Both natural and harmonic minor variants were produced in this manner with the end result that the number of major and minor songs, or more accurately, songs keys, was more equal in number. Ultimately, there were over 3800 instances of songs in major keys and over 3200 in minor keys. Note too, that this was done by only modifying the melody, not the chords. Certainly, this would be inappropriate under most circumstances; however, in this case, the additional complexity associated with translating chords is unnecessary as the key extraction model to be developed is only intended to consider the melody. The chords were not needed and could, thus, be ignored.

4.2 Key Model Creation

The operation of the proposed model, as described earlier, is to accept a window of recently sung notes and produce an indication of which of the two neighboring notes is more likely to be the singer's intended note based on a tonal center inferred from the notes in the window. Specifically, this translated into the development of a model that can take the actual note weights computed from the notes in the window, infer a tonal center based on the windowed notes, and produce a set of note weightings indicative of the note weights that would exist in that tonal center.

This was accomplished by first developing an intermediate model, referred to herein as a "key model" that takes two basic inputs. The first of these inputs is the set of two idealized note distributions, similar to those described in the Krumhansl/Schmuckler algorithm, one each for major and minor keys. The second is an array of 24 key weights, one for each of the possible major and minor keys. These key weights, ranging between 0 and 1, are intended to indicate the degree to which any given key is present in a song's melody. A weight of zero means the absence of that major or minor key. Nonzero entries indicate the relative strength of the presence of that key. This combination of keys is here defined as the "tonal center" referred to earlier.

The output of this model is a distribution of 12 note weights reflective of the given combination of 24 input key weights. To compute that output, the note weights in the idealized distributions are multiplied by the key weight for that particular key, major or minor, after transposing the idealized distribution to the appropriate tonic note for the given key. The total weight for each note is accumulated across all of the 24 keys. The final output note distribution simply consists of that cumulative total for each note. Algorithmically, this is illustrated by the following Python code, with `ndist` being the final note distribution, `params` the "tonal center" distribution, and `distmajor/distminor` the major and minor key distributions:

```
ndist= 12*[0]
for p in range(0, 24):
    if p < 12:
        for i in range(12):
            ndist[(i+p) % 12] += params[p]*distmajor[i]
    else:
        for i in range(12):
            ndist[(i+p) % 12] += params[p]*distminor[i]
```

To avoid large numbers, and produce values that can be more readily interpreted as probabilities, normalization is performed on the final output distribution.

With the key model defined as described, the next task was to extract the 24 key weights for each song in the library. In other words, the tonal center, as defined above, was extracted for each song. The goal in this was to use this extraction process to derive a new set of major and minor note weights for use in this

model, similar to the Krumhansl/Schmuckler weights, but extracted purely from song melodies.

This was accomplished by performing a least-squares fitting of the key model for each song in the library. Least-squares is a standard regression analysis technique used to find a “best fit” set of model parameters for a given set of data. The technique involves minimizing an error function consisting of the sum of the squares of the difference between the model and the data values (referred to as residuals) [15]. The fitting parameters, in this case, were the 24 key weights for the model, which were restricted to positive values. This process requires both an output note distribution against which to fit the model and a set of major and minor note distributions to use as input. The output note distribution was computed directly from the melody for each song. For this purpose, the entire song melody was used, not just a windowed subset.

As a starting point, the major and minor note distributions to be used in the model were extracted from all of the melodies in the library taken in aggregate. Since both the key and the key type, major or minor, for each song are known, distributions for both types were computed by accumulating note usage for the melodies in all library songs of a given type, major or minor, after appropriate transpositions. The result was two distributions, one for major keys and one for minor. Again, the distributions were normalized.

The least-squares fitting of the key model was then performed, as described, for each song resulting in the tonal center for that song using the melody extracted major and minor note distributions. Exercising the resulting model, again with those distributions, produces an output note distribution that, ideally, should correlate well with the original melody distribution. This was, in fact, found to be the case with a mean Pearson correlation coefficient of 0.9125 (standard deviation of 0.055) between the input and output note distributions for all of the, over 7000, songs in the test.

There is a potential consistency issue, however. The input major and minor note distributions were derived directly from the melodies. Even with high correlation, however, the distributions seen in the model output would be expected to differ to some extent from the input distributions, as correlation is not perfect. To minimize this difference, the process described above was iterated. In particular, new major and minor note distributions were derived from the model output, again by summing across all major and minor keys, rather than directly from the melodies. Then, the model was refit using those new distributions instead of the melody-derived input distributions. This process was iterated 25 times, re-deriving the distributions each time, with the final major and minor note distributions ultimately converging to a set of idealized weights. Using these, the mean of the Pearson correlation coefficients between the input and output note distributions for all songs in the library was 0.9291, with a standard deviation of 0.0378 – a slight improvement.

The end result was a set of major and minor note distributions based only on the melodies of a large number of songs and optimized for use in the key model. The final, normalized values are also shown in Table 1, listed under Key model. These distributions can be used to derive a “tonal center” by (Pearson) correlating

them against a distribution derived from a melody for all 24 possible keys, again, retaining only positive values. Then, the key model can be exercised to predict an expected note distribution based on that inferred “tonal center.” As indicated from the results above, the input melody and predicted output distributions should correlate well.

The intention is to use this model on short melody segments rather than on the entire melody. The hope is that high correlation is retained, and the output distribution can be used to predict which notes are more probable. This, in turn, can be used to inform pitch correction. Of course, this hypothesis needed to be tested and verified.

4.3 Note Probability and Confidence

The proposed usage for the key model is as follows. First, a set of recently sung notes is used to compute a note distribution. As described previously, this is then Pearson correlated against the 24 possible major and minor keys using the idealized major/minor distributions, yielding the 24 parameters representing the tonal center for that melody segment. This tonal center is then fed into the key model together with the major/minor idealized distributions. This produces a final output distribution based on the derived “tonal center.”

The notion is to use the weights in the output distribution to determine which of the two closest notes to the actual sung pitch is more likely to be the singer’s intended note. In other words, is the singer sharp or flat? To decide this, the weights of the two notes in question, as determined from the output distribution, are used to compute a “probability” for each with the following simple formulas:

$$P_A = \frac{W_A}{W_A + W_B} \quad P_B = \frac{W_B}{W_A + W_B} \quad (2)$$

W_A and W_B are the model’s output distribution weights for the notes above and below the singer’s current pitch. P_A is to be interpreted as the probability that the note above the current pitch is the correct one, and P_B the probability for the note below. The note with the larger weight, in this way, is taken to be more probable. If the notes have equal weights, they have an equal probability of 0.5.

To test whether this approach actually does provide a useful indication of the correct note, a test was created in which the melodies from the song library were, once again, used. In this test, a window of consecutive notes in the melody was used to construct the input note distribution to the model as discussed. The weight of the next note in the melody (after the window notes) as derived from the model’s output distribution was compared to the weight for an adjacent note, i.e., the one a half step away. The assumption is that the singer is attempting to sing the next melody note, but is sharp or flat, with the actual pitch falling in between the correct melody note

and the adjacent note. If the singer is flat, the adjacent “other” note would be the one a half step below. If sharp, it would be the note above.

In either case, the actual next note in the melody is taken to be the “correct” note, and its probability computed relative to the “other” note as given by the formulas in equations 2 above. If the probability of the actual next note is larger than that of the other note, the model is considered to have correctly selected between the two competing possibilities. The extent to which the model’s prediction is found to be accurate is taken as an indication of how confidently it can be used for pitch correction.

This test was conducted for window sizes ranging between 6 and 32 notes. For each window, approximately 2 to 2.5 million data points were collected, half above and half below the next, correct, note. The data included the weight of the next melody note following the window, as determined by key model, as well as the weights of the “other” notes above and below. Probabilities were computed for the next melody note with respect to those other notes. Overall, it was found that the model “correctly” selected the actual next note for 79.5% of the tested data points using a window size of only 6 notes. This rose to 83% for a window size of 32 notes. Other window sizes between 6 and 32 produced results ranging between these percentages.

It was encouraging that a relatively small window of 6 recent notes produced a predictive capability nearly as good as a larger window of 32 notes, over which one would expect a tonal center to be firmly established. Also, while an ~80/83% accuracy is also encouraging, it still leaves a 1 in 5/6 chance of selecting the wrong note. For that reason, the data were further examined for trends that might provide insight into conditions that might improve predictive capability.

Of particular interest was the possibility that there might be a correlation between strength of the prediction, i.e., the predicted probability, and the overall accuracy. To test this, data were distributed into bins based on the probability of the predicted note, i.e., the larger of the two values. These bins covered a range of probability values between 0.5 and 1.0, as the larger value is always greater than 0.5. Of course, the bins will include both “correct” and “incorrect” predictions, as just defined.

With the data thus separated, the percentage of “correct” predictions within each bin was computed. This is interpreted as a confidence metric for the accuracy of the model’s predictive capability as a function of the probability. Figure 1 plots this relationship, parameterized to show the results for different window sizes, with the confidence (or percent of “correct” predictions) normalized to a maximum value of 1.

The plot clearly shows that the confidence increases with the strength of the prediction/probability. In addition, and as one might expect, confidence is higher the more notes in the window; however, the plots do confirm what was indicated earlier – that smaller window sizes do, indeed, perform nearly as well as larger windows.

The plot in Fig. 2 shows the cumulative percentage of the total tested points above a given confidence level. Again, it is parameterized with the window size. In this case, there is some separation between the window plots with regard to the how

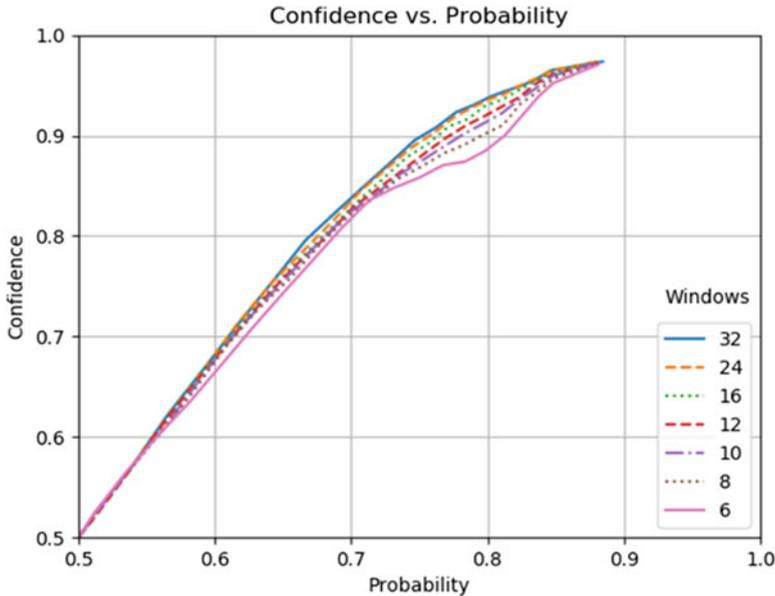


Fig. 1 Key model note selection

frequently higher confidence levels are seen. For example, the arrows indicate the percentages of total points that have 90% or greater confidence for window sizes of 32 and 6 notes. For a window size of 32 notes, 45% of the data have 90% or greater confidence. For a window size of 6 notes, half of that amount, 22.5% show 90% confidence, which indicates that the larger the window, the greater the likelihood of higher probability predictions, as might be expected.

As an aside, it should be noted that attempts were made to look for possible dependencies of prediction accuracy on factors other than the strength of the probability. Among such factors considered were the number of unique notes in the melody window, the actual weight of the selected note, and the strength of the Pearson correlation between the input weight distribution derived from the melody window and the model's output distribution. The prediction accuracy did not indicate a strong dependence on any of these factors. In addition, no significant difference was seen between comparing to the note above as opposed to the note below the actual next melody note.

For comparison, the same analysis was performed using the Krumhansl/Schmuckler weights. The results found were similar, with one significant difference. For the key model, the largest probability, as can be seen in the plot, is around 0.88. With Krumhansl/Schmuckler weights (Fig. 3), the maximum value is only about 0.675. As will become clearer when application of the model is discussed below, this implies that smaller pitch corrections would result were the model to be used for

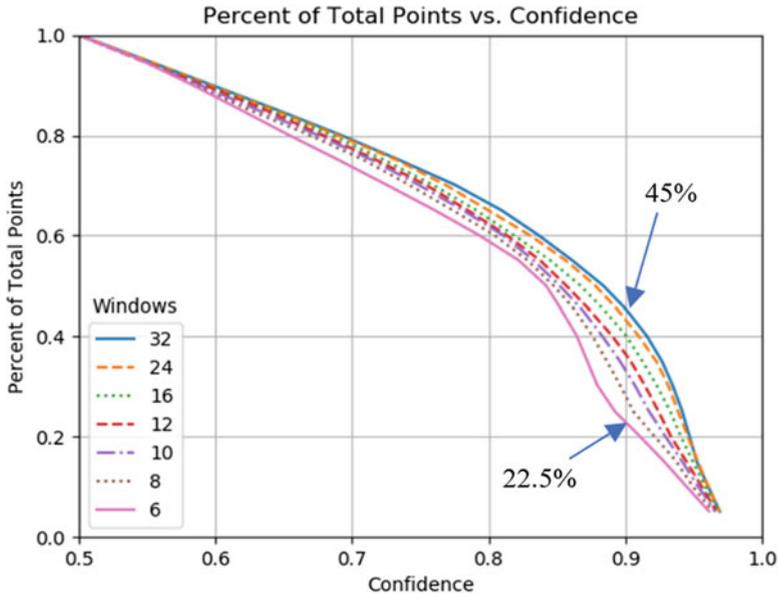


Fig. 2 Key model cumulative confidence

this purpose, indicating, in turn, that the development of melody-only-based weights was a useful exercise.

The data described above were built into a computer model that performs a two-dimensional interpolation to produce a confidence level as a function of input probability and window size. Together with the key model presented earlier, this provides a mechanism for – based on the tonal center inferred from a window of most recently sung notes – predicting which of the two notes closest to a singer’s current pitch is more likely to be the intended note. Specifically, the models provide both an indication of the strength of the prediction, via the probability value, and the confidence in that prediction. In addition, these outputs, whose values range between 0.5 and 1.0, have intuitive interpretation. A probability value of 0.5 indicates that either note is equally probable in the derived tonal center with a value of 1.0 being a firm indication for the selected note. A confidence value of 0.5 indicates a fifty-fifty chance that the prediction is accurate, with a value of 1.0 being absolute confidence.

5 Application

The two models described in the previous sections provide a mechanism for selecting which of the two adjacent notes is more probable in a tonal center derived from a short window of recent notes, as well as an indication of the confidence

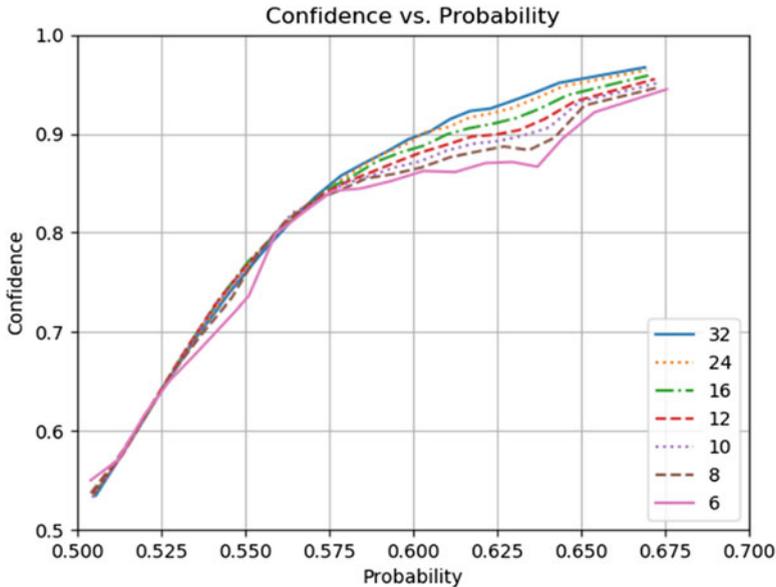


Fig. 3 Krumhansl/Schmuckler note selection

level that one can have in that selection. Generally, the usage model envisioned is to first run the models on a window of recently sung notes. Then, while the next note is being sung, ascertain the singer’s stabilized pitch. From that, determine the two notes nearest to that pitch (i.e., which two notes the pitch is between). Finally, from the key model output, compute the probabilities for those two notes. The note with the larger probability is deemed the likely correct note, with an associated confidence calculated from the confidence model. This can then be used to apply pitch correction while the note is still being sung.

As the original motivation for developing this approach was an existing voice-driven musical instrument, it would be helpful to review the operation of that instrument before addressing the addition of pitch correction to the analysis.

5.1 Voice-Driven Instrument

In the existing voice-driven instrument, the singer’s audio waveform is digitally sampled using a microphone. By default, a standard CD quality sampling rate of 44,100 Hz is used. The waveform data are analyzed in “chunks” of both 1024 and 2048 samples by two separate pitch extraction algorithms [16, 17] using an available Python-based audio library called *aubio* [18]. Details have been described in prior work, and will not be repeated here, except to say that heuristics have been developed to minimize inherent latency and produce a reliable, real time, stream

of the singer's audio pitches. Depending on the platform, and on settings in the application, a new pitch can be produced in the stream in under 4 ms.

Given that pitch stream, additional heuristics were developed to stabilize the pitch as the note develops and determine what musical note is being sung. As mentioned, this can be complicated by fact that even good singers often do not produce a completely accurate and stable pitch. Common issues are inaccurate overall pitch, inaccurate attack (note onset), pitch drift as the note develops, and vibrato.

Since the goal is live performance, one does not have the luxury of waiting for the pitch to stabilize, assuming it does adequately. The approach taken was to have the targeted instrument follow the singer's actual pitch stream at onset and then migrate to a more stable, median value as the note develops. A balance must be struck between responsiveness and stabilization and, again, this is built into the heuristics.

Complicating this further is the possible presence of vibrato, which is a low-frequency oscillation, usually between 4 and 8 Hz, superimposed on the pitch stream. Depending on the singer, this can be quite pronounced, with the amplitude easily extending to nearby notes. While this is generally perceived as appealing in singing, it needs to be removed, or averaged out, for the conversion task being attempted in the voice-driven instrument as the vibrato sound may not be appropriate for the target output instrument.

Once a stable pitch stream has been generated, the final step in the process is musical note recognition. The approach used was to compare the most recent pitch frequencies in the stabilized pitch stream to the closest musical note frequencies. At any point in time, a given stabilized pitch will generally be between the frequencies of two musical notes. Over a window of recent time, a score is generated for the nearby musical notes which considers the fraction of total pitches that are closest to that note, weighted by how close the pitch is to the note. When a musical note attains a high enough score, the output frequency, i.e., the frequency that the virtual instrument is directed to play, "locks on" to the frequency of the musical note if it is close enough. If neither note is dominant, the stabilized frequency is played instead – basically following what the singer is singing.

The inherent assumption in this note recognition is that both of the nearest notes are considered equally likely, with the distance from each treated equally in determining "closeness." It is in this that an opportunity for pitch correction can be found.

5.2 *Pitch Correction*

When deciding which musical note should be played, one considers both the frequencies of the musical notes nearest the actual pitch frequency and the distance from those notes. If the nearest notes are equally probable, and the pitch frequency is exactly midway between the note frequencies, no decision can be made regarding

which note to “lock onto.” On the other hand, the closer the pitch is to either note, the greater the confidence in that note.

The suggested approach to pitch correction is to skew the balance point between the two candidate notes, based on the modeled probability of the two notes. Without pitch correction, (equal note probability) any pitch that falls above the midpoint between note frequencies would add to the previously discussed note score for the upper note. A pitch that falls below the midpoint would increase the score for the lower note. The balance point is at the midpoint. If, however, the upper note was deemed to be twice as probable, the balance point would move downward so that any pitch that falls in the upper 2/3 of the range between the notes would add to the score of the upper note, with pitches in the lower 1/3 of the range contributing to the lower note score.

This approach of adjusting the balance point does not preclude the singer from producing a note which is less probable, based on tonal center analysis. It simply biases the output toward the more probable note. If the singer accurately sings the less probable note, its score will still be highest and it will get played. This is necessary because, as indicated earlier, the model cannot predict the correct note with absolute certainty. Even in the 22.5–45% of the cases where the modeled confidence level is 90% or more, there is still as much as a one in ten chances that the wrong note will be selected. To mitigate against this, the adjustment to the balance point can take both the probability and confidence values into account.

One possible adjustment is to scale the probability with the confidence value. For example, if the probability is 0.8 with a confidence of 0.9, the scaled probability would be 0.77, computed as follows (since the probability cannot be below 50%):

$$P_{\text{scaled}} = 0.5 + \text{confidence} * (P_{\text{computed}} - 0.5) \quad (3)$$

Additionally, in the existing algorithm, a requirement is imposed that the stabilized pitch be within a certain range of a target note. This range can also scale with movement of the balance point. This has been implemented with some preliminary testing in the voice-controlled instrument.

An example illustrating the result is shown in Fig. 4. The figure shows the singer’s pitch as a function of time, which is given in milliseconds. It focuses on a note 32 seconds (32,000 ms) into the song, the duration of which is a bit under 2 seconds (ending at about 33,860 ms). The dotted line, labeled “micpitch” is the singer’s pitch as captured from the microphone. It shows oscillation due to vibrato. The dashed line, labeled “stablepitch,” is the output of the pitch stabilization algorithms described earlier. As can be seen in the plot, this initially follows the singer’s pitch, moving to a median value as the note develops. At approximately 32,700 ms, the vibrato is detected, as indicated in the first of the subplots. The stabilized pitch then becomes the center pitch of the vibrato oscillation rather than the median pitch value.

The solid line, labeled “pitch,” is the final output pitch that would be produced if no pitch correction was done. It is the result of some additional smoothing of “stablepitch.” Note that it closely follows the stable pitch line, never locking on to the C4 note, even though it is closer to C4 than to B3. This is due to the fact that the

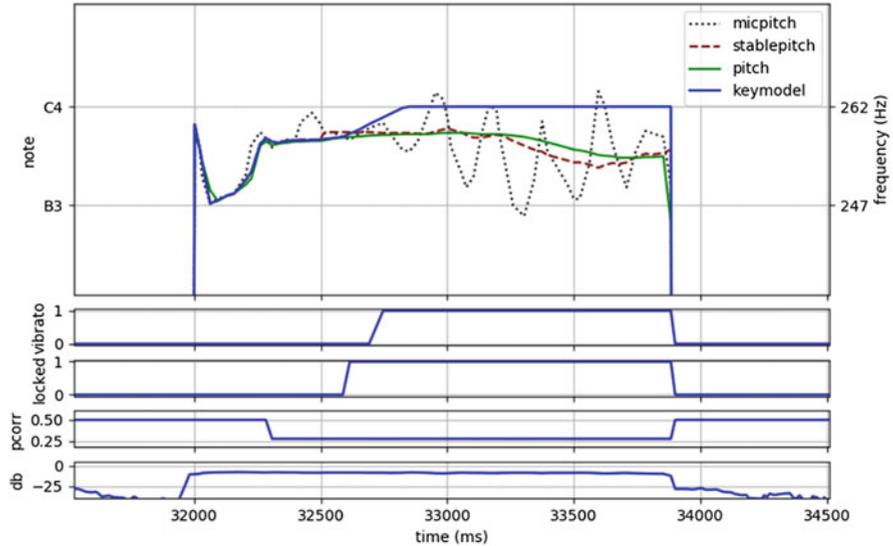


Fig. 4 Pitch correction

stable pitch is just outside of the range set in the algorithm for note locking to occur (200 cents, in this case, where 1 cent is 1/100th of the musical half step between two adjacent notes, B3 and C4 in this case).

The actual output pitch is represented by the remaining solid line, labeled “keymodel.” In this case, the output locks on to C4, as can be seen in both the pitch itself and in the second subplot (“locked”) that indicates a locking event has occurred. To understand this, consider the third subplot, labeled “pcorr” for pitch correction. This shows the movement of the balance point discussed earlier. After the initial pitch stabilization time, the balance point moves from 0.5, the point midway between the C4 and B3 frequencies, down to 0.276. While not shown on the plot, this is the result of a calculated probability of 0.75 from the key model, with an attendant confidence of 0.896, over a window of 30 notes – meaning that C4 is approximately three times more likely than B3 in this inferred tonal center. This not only makes more of the stable pitches contribute to the score for C4, but it also widens the locking range for the C4 note (by about 45%), and results in the observed note lock. (The final subplot, “db,” is the volume level, which is used to trigger a note event.)

6 Results

Testing the presented model is complicated by the fact that it requires collecting data from actual, real time, use of the voice-driven instrument, and judgments regarding efficacy can be subjective. Despite those complications, the following test results have been generated and can be reported upon.

For the test, singers were asked to sing melodies a cappella, i.e., with no accompaniment. The instrument has the capability of capturing and storing the pitch stream from the microphone. This pitch stream was then played back into the instrument (another capability) twice. The first time, no pitch correction algorithm was applied. The instrument simply processed the pitch stream and drove the virtual instrument as described earlier. The second time, pitch correction was applied using the key model. In both cases, the output pitch stream was captured and analyzed.

The analysis consisted of performing a comparison of the correct notes in the melody to the stabilized and, where applicable, “locked on” pitches produced by the algorithm. The goal was to ascertain whether the pitch correction algorithm produced an improvement to the final output as compared to the base algorithm without pitch correction. Thus, the analysis is focused on comparing the output of the correction algorithm to the algorithm without correction.

As mentioned, there is a certain amount of subjectivity in any such analysis. The considerations used here were as follows. The pitch from an analysis was compared to the correct melody note to determine if the analyses, both with and without correction, agree with each other and/or with the correct note. Additionally, it was determined whether either analysis “locked on” to a note, and if so, did the locking duration increase or decrease with the addition of pitch correction? An increased locking duration would mean that the locking occurred earlier, with the converse true if it decreased.

To perform this analysis, a simple program was written that sequentially examined each note in the song. Using the considerations described, it came to a decision as to whether the application of pitch correction improved that note’s recognition. For example, did a previously unlocked note (in the algorithm without correction) become locked when correction was applied? If it did, and it locked to the correct note, that would be an improvement. If the reverse were true, or it locked to the wrong note, it would imply a degradation in the performance. If the locked time for a correct note increased, with respect to a note that was locked without correction, that would also be an improvement, with the converse, again, being a degradation. If there was no significant difference in the output, the conclusion is that there is neither improvement nor degradation when applying the correction algorithm.

Using this approach, results have been generated for a small number of example songs performed by two experienced professional singers, one man and one woman. These are summarized in Table 2.

Each row in the upper portion of the table shows data for a separate song, five from the first singer and four from the second. The first column indicates the song, by number, and singer, also by number (1 or 2). All were popular tunes

selected by the singer (there was only one common song between the two singers). Each song was sung straight through and contained the number of analyzed notes shown in the *Number of notes* column. Notes that were too short in duration (under 200 ms) for a lock to occur were ignored in the analysis – approximately 6% overall. Pitch correction, when it was enabled, began when the window size reached 6 and continued with the window increasing in size for each note until the model maximum of 32 was reached, at which point the last 32 notes were used for subsequent corrections. The initial 6 notes were also ignored, as no pitch correction was done until the minimum window size of 6 was reached.

The analysis described above was performed on potentially affected notes, where the correction algorithm was compared to the no-pitch-correction case, and results are displayed. The number of notes indicating an improvement for a given song is in the *Better* column, and the number showing degradation is in the *Worse* column. Those labeled *Neither* were judged to have no significant change. The total number of analyzed notes for each song is in the last column *Number of notes*. The grand total of analyzed notes, 484, is at the bottom of that column.

The row labeled *Total* indicates the number of *Better* (82) and *Worse* (19) notes for all songs, for a total of 101 affected notes. Thus, the *Percent of affected* notes (next row) that were made better was 81% (82/101), with 19% (19/101) made worse. The last row, labeled *Percent of total*, shows the percentages of the 484 total analyzed notes that were made *Better* and *Worse*, as well as those that were unaffected (*Neither*).

From the *Percent of total* values, it can be seen that, in most cases, 78%, the pitch correction algorithm did not measurably affect the results. In the 22% of the cases where it did affect a measurable change, the change was an improvement for 81% of those notes (17% overall), and a degradation for 19% (4% overall), which is a ratio of over 4.25 to 1 (81/19) in favor of improvement.

Table 2 Pitch correction statistics

Song/singer	Better	Worse	Neither	Number of notes
1/1	8	5	45	58
2/1	5	1	11	17
3/1	17	4	70	91
4/1	11	2	48	61
5/1	14	5	86	105
1/2	8	0	40	48
2/2	12	0	35	47
3/2	4	1	12	17
4/2	3	1	36	40
Totals	82	19		484
Percent of affected	81%	19%	383	
Percent of total	17%	4%	78%	

It is important to note that, while it is not shown in the table, no cases were seen in which the algorithm “corrected” to a wrong note. In other words, a note that was correctly played without pitch correction enabled was never adjusted to a wrong note by the model. The converse was also true; an incorrect note was never adjusted to a correct note by the algorithm.

The results indicate that, in all cases, the performance improved, albeit marginally. For most notes, 78% of them, there was no measurable impact at all. Of course, for experienced singers, this should not be surprising. One would expect a professional singer to produce reasonably accurate notes.

It is in the remaining portion of the performance, where the singer is less accurate, i.e., a bit sharp or flat, that a benefit is hoped for, and for which the algorithm is designed. Indeed, by an over 4.25 to 1 margin, improvements do, indeed, occur for such cases. As mentioned, no case was observed where the application of the algorithm generated a spurious note, so any fears of “correcting” to a wrong note were, at least for these examples, seen to be unfounded.

Overall, the net effect is that fine-tuning of a performance occurs, rather than a drastic change. While it cannot be shown here, audible differences are noticeable, but subtle as well – a general sense of improved tuning, and no obvious “bad notes” resulting from the pitch correction.

7 Conclusions/Additional Directions

A model for applying pitch correction in a live musical performance was presented. The model is targeted specifically for use in an application wherein a singer is, in effect, playing a separate virtual musical instrument with their voice.

The model extracts a “tonal center” from recently sung notes and uses that information to determine which note is more likely in cases where the singer is flat or sharp. A byproduct of the effort is a new set of note distribution weights for major and minor keys based purely on a song’s melody. The model produces both a probability for the more likely note and a confidence metric for its accuracy. Use of the model for pitch correction as well as its integration into an existing voice-driven instrument was presented.

Test results generated from the actual use of the instrument indicated improvement in the output pitch when correction was applied. For the target user, a reasonably experienced singer who wishes to “play” another instrument with their voice, these results support the contention that pitch correction, as applied by the model, could be a useful performance aid.

Future work is focusing on two areas. First is more testing using, again, singers in live performances to collect additional data. Listener perception tests and analysis as a function of window size are also areas of interest, as would be a comparison of the presented technique, wherein a tonal center is inferred, against one in which the song key is set by the user.

Second is a parallel effort that is currently underway to produce a model based not on tonal analysis, as was done here, but on a neural network that is being trained directly using melody data. Promising results are being seen in that approach as well and will be reported upon in a separate paper.

References

1. J. Carelli, Voice to musical instrument translation in a performance environment, in *Proceedings of the 2019 International Conference on Software Engineering Research & Practice (SERP'19)*, Las Vegas, Nevada, pp. 54-60, July 29 – August 1, 2019
2. P. Manning, *The Development of the MIDI Communications Protocol: from Electronic and Computer Music* (Oxford University Press, New York, 2013)
3. P. Larrouy-Maestri, D. Magis, D. Morsomme, The evaluation of vocal pitch accuracy: The case of operatic singing voices. *Music Percept. Interdiscip. J.* **32**(1), 1–10 (2014)
4. G. Verena Sampaio de Souza, J.M.T. Duarte, F. Viegas, M. Simoes-Zenari, K. Nemr, An acoustic examination of pitch variation in soprano singing, *J. Voice* (2019). <https://doi.org/10.1016/j.jvoice.2018.12.007>
5. www.celemony.com/en/melodyne/what-is-melodyne, Celemony Melodyne web site, verified 3/15/2020
6. www.antarestech.com, Antares Autotune web site, verified 3/15/2020
7. C.L. Krumhansl, *Cognitive Foundations of Musical Pitch* (Oxford University Press, New York, 1990)
8. en.wikipedia.org/wiki/Pearson_correlation_coefficient, Pearson Correlation Coefficient (Wikipedia), verified 3/15/20
9. Muscorescore, <https://musescore.org/en>, verified 4/24/20
10. M.S. Cuthbert, C. Ariza. music21: A toolkit for computer-aided musicology and symbolic music data, eds. by J. Stephen Downie, R. C. Veltkamp, in *11th International Society for Music Information Retrieval Conference (ISMIR 2010)*, August 9–13, Utrecht, Netherlands (2010). pp. 637–642
11. C.L. Krumhansl, E.J. Kessler, Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys. *Psychol. Rev.* **89**, 334–368 (1982)
12. H. Bellmann, About the determination of key of a musical excerpt, in *Computer Music Modeling and Retrieval. CMMR 2005. Lecture Notes in Computer Science*, ed. by R. Kronland-Martinet, T. Voinier, S. Ystad, vol. 3902, (Springer, Berlin, Heidelberg, 2006)
13. D. Temperley, *The Cognition of Basic Musical Structures* (MIT Press, Cambridge, 2001)., ISBN 978-0-262-20134-6
14. B.J. Aarden, Dynamic melodic expectancy, Doctoral Dissertation, The Ohio State University, 2003
15. https://en.wikipedia.org/wiki/Least_squares, Least Squares (Wikipedia), verified 3/21/20
16. A. de Cheveigne, H. Kawahara, YIN, a fundamental frequency estimator for speech and music. *Acoust. Soc. Am.* **111**(4), 1917–1929 (2002)
17. O. Babacan, T. Drugman, N. d' Alessandro, N. Henrich, T. Dutoit, A comparative study of pitch extraction algorithms on a large variety of singing sounds, in *38th International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2013)*, Vancouver, Canada (2013). pp.1–5
18. P.M. Brossier, Automatic annotation of musical audio for interactive applications, Ph.D Thesis, Centre for Digital Music, Queen Mary University of London, March 2007

Analysis of Bug Types of Textbook Code with Open-Source Software



Young Lee and Jeong Yang

1 Introduction

The importance of example code cannot be overemphasized, and students take the source code in the textbook as a standard of exemplary code. As far as we know, we have not given enough attention or efforts to choose these example codes. In this paper, we studied the quality of textbook sample codes by detecting bugs with bug types and compared them with the bug types detected in the Open-Source Projects. As examples play the most helpful resource in teaching and learning programming [1], student programmers use the code examples as templates for their own work, and important code examples are provided by textbooks [2]. Examples must therefore should not exhibit any undesirable properties or behavior. Code examples with errors may confuse students to learn in developing correct programs [3].

This study provides empirical evidence on the quality of textbook code examples by analyzing bug types and comparing them with those in Open-Source Projects. The main research questions are as follows: (1) Do textbooks use the quality source code examples? This question will be answered by analyzing code examples using static code analysis tools, and (2) how do static analysis results relate to OSS? Hence, the following sub-questions will be investigated:

SQ1. Are textbook code examples correct and bug-free?

SQ2. Which bug types are the most occurred in textbooks?

Y. Lee (✉) · J. Yang

Department of Computing and Cyber Security, Texas A&M University–San Antonio, San Antonio, TX, USA

e-mail: young.lee@tamusa.edu; jeong.yang@tamusa.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_44

629

SQ3. Are the bug types in textbooks also found in the OSS?

SQ4. Are the bug types in OSS also found in the textbooks?

2 Related Work and Background

2.1 Textbook Code Examples

One critical point in recent literature is the integration of object-oriented paradigms in the instruction of programming. The work of Nordström et al. discussed the flaws of common textbook examples and how to improve the quality of examples. Their study revealed that the object-oriented quality of examples is low [4]. A number of scholars noted that introductory programming courses using the object-oriented paradigm are more complicated, compared to the imperative/procedural paradigm [5–7]. When the object-oriented concepts are discussed, examples are important for learning [8–11], because, in the educational context, examples must be easy to understand for learners, but still exemplary to act as role models for the paradigm.

Textbooks can be a major source for examples of common programming problems in introductory programming courses. Many textbook examples have been evaluated through a large-scale study to capture technical, didactical, and object-oriented qualities [12]. The particular needs of a novice being introduced to object orientation were considered, and some heuristics for the design of object-oriented examples for novices were developed from those. The discussion for teaching object orientation with examples is also initiated in [4], and the design of examples is specifically discussed [13].

2.2 Static Code Analysis

Static code analysis is the process by which software developers review and examine their code for problems and inconsistencies. Static code analysis tests source code through scanning without executing, but after compiling. The source code review is critical to enhancing software security through structured design, code inspection, and peer review of the code. It can be integrated into the software development process to help developers detect potential vulnerabilities at the early stage of the development, reducing risks prior to a production environment.

The code analysis can be done using static code analyzers – tools to assist in identification of security vulnerabilities, which developers can use in examining and analyzing their source code. Such example tools are FindBugs, Find Security Bugs, Fortify, PMD, Lapse+, and SCALe. These tools examine the code and automatically detect potential errors and bugs that pass through a compiler. FindBugs is a static code analysis tool for Java programs [14] and highly configurable tool that

allows loading custom rulesets. The customizable rulesets can detect typical errors including security-related checks. In a recent study, Oskouei et al. used three well-known open-source bug-finding tools, PMD, FindBugs, and Checkstyle, to run and compare results on a variety of open-source Java programs.

3 Research Methods

3.1 Code Analyzer Tools

When FindBugs is in action in analyzing source code, its reporting categorizes bugs to bug types: Bad Practice, Correctness, Malicious Code Vulnerability, Performance, Security, Dodgy Code, Multithreaded Correctness, Experimental, and Internationalization.

Bad Practice (B) code violates recommended and essential coding practices. The examples of Bad Practice include equals problems, improperly formatted strings, dropped exceptions, serializable problems, and misuse of finalizing. *Dodgy Code (D)* is a confusing code that is anomalous or written in a way that can lead to errors. Examples include dead local stores, unconfirmed casts, division overflows, useless object creation, and switch fall through, unconfirmed casts. *Correctness Bugs (C)* are probable bugs with apparent coding mistakes that are probably not what developers intended. They can produce unwanted results.

Performance (P)-related inefficient code can cause performance degradation and resource wasted. For example, when a class contains an instance final field that is initialized to a compile-time static value, it should be considered to be a static field. Unread fields that are never read can be removed from the class. *Experimental (E)* code can miss cleanup of streams, database objects, or other objects that require a cleanup operation. For example, a method may fail to clean up (close, dispose of) a stream, database object, or other resource requiring an explicit cleanup operation. *Internationalization (I)* code can inhibit the use of international characters. Using a default encoding can lead to incompatibility on systems with certain defaults. For example, when the default encoding is used for the scanner input, the use of utf-8 can resolve the issue since the presence of “utf-8” explicitly declares the encoding of the scanner. *Security (S)* includes Multithreaded Correctness, Malicious Code Vulnerability, Predictable Random, Potential Path Traversal, and other security-related bugs.

3.2 Bug Data Collection

Text 1 – Starting out with Java: From Control Structures through Objects [15] was used to analyze the source code as it represents the most modern example

of beginner Java concepts being taught across colleges and universities. Text 2 – Introduction to Java Programming and Data Structures, Comprehensive Version [16], was also used for the analysis. As the goal was to look for the presence of bugs, when analyzing the code, the data was classified and collected into different categories.

The analysis was conducted with the Java code examples throughout the two textbooks. As the title of the books indicate, while Text 1 covers fundamental Java concepts from control structures to objects, and beyond, Text 2 covers data structures concepts as well as the fundamentals. It should be noted that there were intentionally incomplete code examples.

Liu et al. claim that those violations that are recurrently fixed are likely to be true positives, and an automated approach can learn to repair similar unseen violations [17]. Liu et al. collected and tracked a large number of fixed and unfixed violations across revisions of Open-Source Software and provided insights into prioritizing bug types. Data collection of OSS was based on earlier work by [17]. In the present work, we have compared the dataset of OSS with dataset of Textbooks. We referenced bug data from the real-world Open-Source Java Projects used in the study [17, 18]. These bug datasets are from 730 projects and detected 400 violation types and 16,918,530 distinct violations.

4 Research Results

4.1 Source Code in Texts

For the textbook code examples, the groups were composed based on similar topics of the source code in the consecutive chapters as presented in Table 1. The basic groups represent contents from both Text 1 and Text 2, while the advanced groups represent contents from Text 2, which is a comprehensive version covering the concepts of data structures and more. We consider that modern examples of Java concepts are being used in the books and taught across colleges and universities.

Table 1 Texts group composition

Group no	Basic groups	Group no	Advanced groups
1	Fundamentals, control structures, methods	7	Fundamentals data structures
2	Arrays	8	Algorithms
3	OOP	9	Trees
4	File I/O	10	Graphs
5	Recursion	11	Collection streams
6	Databases	12	Networking and parallel
		13	Internalization

Table 2 Bugs in category for basic groups in both texts

Gr.	I	D	B	C	P	E	S	Total
1	65	4	16	0	8	0	7	100
2	15	4	2	0	0	0	1	22
3	17	5	15	5	10	0	8	60
4	24	8	9	2	2	0	2	47
5	5	5	0	0	2	0	0	3
6	4	0	13	0	3	30	14	64
Total	130	26	55	7	25	30	32	305

Table 3 Bugs in category for advanced groups in Text 2

Gr.	I	D	B	C	P	E	S	Total
7	2	1	5	0	3	0	0	11
8	6	0	4	0	2	0	0	12
9	1	1	3	0	1	0	0	6
10	2	0	8	0	8	0	3	21
11	4	0	2	0	0	0	0	6
12	0	0	0	0	16	0	0	16
13	0	0	0	0	4	0	0	4
Total	15	2	22	0	34	0	3	76

Table 4 Bug rates in basic groups

Gr.	Total # of bugs	# of files with bugs	# of files scanned	Bug rate
1	100	79	151	52.3%
2	22	15	49	30.6%
3	60	47	104	45.2%
4	47	30	48	62.5%
5	3	9	35	25.7%
6	64	18	25	72%
Total	305	198	412	48.1%

A total of 227 files were scanned throughout Text 1 with 16 chapters on the 6 basic groups, and 303 files were scanned throughout Text 2 with 32 chapters on the 7 advanced groups. Table 2 shows the total number of bugs found in each bug category from both Texts for the 6 basic groups, and Table 3 shows the total number of bugs found in each bug category from Text 2 for the 7 advanced groups.

The analysis results indicate that 38% (145 out of 381) of the bugs found are internalization bugs. Group 1 has the most bugs.

As presented in Table 4, group 6 has a significantly larger bug rate, 72%: 18 out of 25 scanned files, compared to other groups, while the first group has the most bugs with a relatively large number of files scanned. The large portion of the bugs in group 4 varied from simple bugs such as Bad Practice or Correctness, which ranged from using default encoding like the Scanner class to implementation issues of methods being used. The bugs discovered attributed to most of the weight when computing the bug rate, which resulted in the second-highest rate (62.5%).

Table 5 Bug rates in advanced groups

Gr.	Total # of bugs	# of files with bugs	# of files scanned	Bug rate
7	11	9	29	31.1%
8	12	9	21	42.9%
9	6	2	10	20.0%
10	21	9	19	47.4%
11	6	6	12	50.0%
12	20	9	28	33.0%
Total	76	44	119	36.97%

Groups 2 and 5 had relatively small bug rates at 30.6% and 25.7%, respectively. Upon examining group 3, the number of bugs found in Text 1 was far more surmountable than the bugs found in Text 2. Despite having more or fewer bugs, the texts shared a commonality with the types of bugs discovered.

Overall, Text 1 revealed a relatively moderate bug rate of 44.42%, and Text 2 produced a slightly larger bug rate of 53.80%, together 48.1%. The most interest in the results is that 77.36% (41 out of 53) of the elementary programming source code in group 1 of Text 2 contains 100 bugs (93 regular bugs and 7 security bugs). The second most interested group is that 70% (7 out of 10) of the database-related source code in group 6 of Text 1 contains 25 bugs. In summary, 305 bugs were found from 198 Java files out of 412 files scanned.

The advanced topics covered in Text 2 were classified into a multitude of groupings with corresponding chapters. Several subgroups were formed covering an extensive amount of data structures and algorithms such as trees, graphs, and sorting algorithms. Most of these topics are appropriate to be taught in CS2 and/or data structures courses. While the groups can be examined individually, they were analyzed as an overall group for advanced topics (Table 5).

As presented in Table 3, the analysis results indicate that 44.7% (34 out of 76) of the bugs found are performance and inefficient code-related bugs in the regular category. While these trends are different from the bug findings for the beginner Java concepts, Bad Practice also consistently presents throughout the chapters in Text 2. Group 11 had a larger bug rate, 50%: 6 out of 12 scanned files, compared to other groups, with a close bug rate 47.4% in group 10.

4.2 With Open-Source Project Files

Table 6 compares the bug distribution rate of each of the bug categories on Texts and OSS. The data shows that 42.6% of the bugs found in Texts relate to Internationalization, while the Open-Source Projects have 39.8% of their source code associated with Dodgy Code. Interestingly, Bad Practice issues consistently present in both Text (18.0%) and OSS (26.4%) groups.

Table 6 Bug distribution rate (%) in Texts and OSS

Category	I	D	B	C	P	E	S
Textbooks	42.6	8.5	18.0	2.3	8.2	9.8	10.5
OSS	4.4	39.8	26.4	3.2	10.8	0.8	16.6

Table 7 Top 20 bug types detected in Texts

Text rank	C.	Bug type detected	No.	OSS rank
1	I	DM_DEFAULT_ENCODING	144	10
2	E	OBL_UNSATISFIED_OBLIGATION	31	42
3	P	DM_NUMBER_CTOR	25	26
4	D	DLS_DEAD_LOCAL_STORE	15	4
5	B	VA_FORMAT_STRING_USES_NEWLINE	14	68
5	S	DMI_CONST_DB_PW/SQL_NONCONS_ST_PASSED_TO_EX	14	291
7	C	VA_FORMAT_STRING_ILLEGAL	11	360
8	B	ODR_OPEN_DATABASE_RESOURCE	9	77
8	P	URF_UNREAD_FIELD	9	118
8	P	DM_NEXTTINT_VIA_NEXTDOUBLE	9	215
11	B	HE_EQUALS_USE_HASHCODE	8	50
11	P	SIC_INNER_SHOULD_BE_STATIC_NEEDS_THIS	8	103
11	B	NM_CLASS_NOT_EXCEPTION	8	201
14	P	SS_SHOULD_BE_STATIC	7	53
14	C	RV_ABSOLUTE_VALUE_OF_RANDOM_INT	7	229
14	C	SF_DEAD_STORE_DUE_TO_SWITCH_FALLTHR_TO_THR	7	382
17	S	PT_ABSOLUTE_PATH_TRAVERSAL	6	323
18	M	MS_EXPOSE_REP	5	134
19	P	SBSC_USE_STRINGBUFFER_CONCATENATION	4	45
19	C	EQ_SELF_USE_OBJECT	4	102

Table 7 provides the distribution of the top 20 bug violation types detected in the code examples in the two textbooks. These are ranked based on their violation occurrences. It was observed that 11 bug types out of the top 20 are also ranked 103rd or higher in the distributions of bug violation type rankings for the OSS (see Table 7). Standard Bug patterns and each of their bug types reported by FindBugs and SpotBugs are described in [18, 19].

DM_DEFAULT_ENCODING ranked first with 144 instances, which is 39.3% of all instances (144 out of 367) found. This bug is ranked tenth in the OSS group. DM_DEFAULT_ENCODING is categorized in Internationalization which code flaws having to do with internationalization. It indicates a call to a method which performs a byte to string conversion and vice versa and assumes that the default platform encoding is suitable. This may cause various application behaviors among platforms. To fix DM_DEFAULT_ENCODING violation, an alternative API should be used explicitly.

OBL_UNSATISFIED_OBLIGATION ranked 2nd with 31 instances and ranked 42nd in the OSS. OBL_UNSATISFIED_OBLIGATION is categorized

in Experimental. This violation is detected when a method fails to clean up a stream, database object, or other resource requiring an explicit cleanup operation. `DM_NUMBER_CTOR` ranked 3rd with 15 instances and ranked 26th in the OSS. This violation is categorized in Performance, and it recommends to use either autoboxing or the `valueOf()` method when creating instances of Long, Integer, Short, Character, and Byte.

`DLS_DEAD_LOCAL_STORE` ranked fourth with 15 instances and also ranked fourth in the OSS. `DLS_DEAD_LOCAL_STORE` is categorized in Dodge Code, which is detected when a local valuable has an assigned value without being used in any of subsequent code instruction. Java compiler also detects this violation with final local variables as FindBugs and SpotBugs tools detect bugs from byte code after compilation. However, there is no known way to eliminate these false positives [19]. Due to this reason, it is not certain how much percentages of the detected `DLS_DEAD_LOCAL_STORE` instances are the actual instances or false positives.

`VA_FORMAT_STRING_USES_NEWLINE` ranked 5th with 14 instances and ranked 68th in the OSS. This type is Bad Practice. It detects the format string includes a newline character (`\n`). In format strings, it is generally preferable to use `%n`, which will produce the platform-specific line separator.

`VA_FORMAT_STRING_ILLEGAL` ranked 7th with 11 instances and ranked 360th in the OSS. This bug type is categorized in Correctness. It detects the format string syntactically invalid and causes a runtime exception. It was observed that this violation type appears frequently in the textbook code examples but rarely occurred in the OSS.

`SF_DEAD_STORE_DUE_TO_SWITCH_FALLTHROUGH_TO_THROW` ranked 14th with seven instances and ranked 382th in the OSS. This violation type is categorized in Correctness. It detects when a value stored in the previous switch case is ignored due to a switch fall through to a place where an exception is thrown. It is likely caused by missing a break or return at the end of the cases.

Table 8 provides the distribution of the top 20 bug violation types detected in the OSS. These are also ranked based on their violation occurrences. `SE_NO_SERIALVERSIONID` ranked first with 1,385,971 instances, which is 8.2% of all 16,918,530 instances; however, this bug type is not found in the code examples of the textbooks. `SE_NO_SERIALVERSIONID` is categorized in Bad Practice. The class with this type implements a Serializable interface, but doesn't define a `serialVersionUID` field. To ensure interoperability of the Serializable interface across versions, an explicit `serialVersionUID` must be added.

`RCN_REDUNDANT_NULLCHECK_OF_NONNULL_VALUE` ranked second but not found in the textbooks. This bug type is categorized in Dodge Code, which is detected when a method contains a redundant check of a known non-null value against the constant null. `BC_UNCONFIRMED_CAST` ranked third but not found in textbooks. This bug type is categorized in Dodge Code, which is detected when the cast is unchecked. Not all instances of this type casted from can be casted to the type it is being cast to. To prevent this violation type, the program logic must be checked to ensure the cast will not fail.

Table 8 Top 20 bug types in OSS

OSS rank	C.	Bug type detected	Count	Text rank
1	B	SE_NO_SERIALVERSIONID	1385971	x
2	D	RCN_REDUNDANT_NULLCHECK_OF_NONNULL_VALUE	870591	x
3	D	BC_UNCONFIRMED_CAST	716810	x
4	D	DLS_DEAD_LOCAL_STORE	615898	4
5	M	EI_EXPOSE_REP	540693	x
6	P	SIC_INNER_SHOULD_BE_STATIC_ANON	532371	x
7	M	EI_EXPOSE_REP2	530962	x
8	B	SE_BAD_FIELD	514687	x
9	B	NM_METHOD_NAMING_CONVENTION	468914	x
10	I	DM_DEFAULT_ENCODING	421185	1
11	D	REC_CATCH_EXCEPTION	411085	x
12	D	UWF_FIELD_NOT_INITIALIZED_IN_CONSTRUCTOR	386809	x
13	D	PZLA_PREFER_ZERO_LENGTH_ARRAYS	370524	x
14	D	BC_UNCONFIRMED_CAST_OF_RETURN_VALUE	341272	x
15	D	RI_REDUNDANT_INTERFACES	323661	x
16	I	DM_CONVERT_CASE	319207	x
17	D	ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD	307155	24
18	D	SF_SWITCH_NO_DEFAULT	305826	x
19	M	MS_SHOULD_BE_FINAL	291123	x
20	D	NP_LOAD_OF_KNOWN_NULL_VALUE	236142	x

EI_EXPOSE_REP and EI_EXPOSE_REP2 ranked fifth and seventh, respectively, but not found in the textbooks. These violation types are categorized in Malicious Code Vulnerability. They are detected when a reference to a mutable object value stored in one of the object’s fields. If instances are accessed by untrusted code, unchecked changes to the mutable object would compromise security or other important properties. Returning a new copy of the object is a better approach in many situations.

Only three violation types (DLS_DEAD_LOCAL_STORE, DM_DEFAULT_ENCODING, ST_WRITE_TO_STATIC_FROM_INSTANCE_METHOD) from the top 20 were found in the code examples in the Texts. In summary, 400 different bug types are found at least once in the OSS, and 36 different bug types are found in the textbooks.

5 Conclusion and Future Work

Using static analysis tools, FindBugs and SpotBugs, this study analyzed the code examples of two widely adopted collegiate level programming textbooks and compared the bug types from textbook code examples and open-source Java

software. This study aims to analyze the code examples of two textbooks using static analysis tools. The study thoroughly analyzed the code examples to detect bugs and compared them with the bugs found in real-world Open-Source Projects to ensure the quality of code examples used in the textbooks. Overall, 42.6% of the bugs found in the Texts relate to Internationalization, while the Open-Source Software (OSS) has 39.8% of their source code associated with Dodgy Code. Bad Practice issues consistently present in both Texts (18.0%) and OSS (26.4%) groups. DM_DEFAULT_ENCODING violation type was detected the most with 39.3% of all instances found in Texts. SE_NO_SERIALVERSIONID ranked first with 8.2% of all instances in OSS; however, this bug type is not found in any of the textbook code examples. DLS_DEAD_LOCAL_STORE in a Dodge Code category ranked fourth in both Texts and OSS. Textbooks are missing certain code examples that are related to the high-ranked bug types in OSS.

The top four bug types in OSS not found in the textbooks are URF_UNREAD_FIELD (B), RV_RETURN_VALUE_IGNORED_NO_SIDE_EFFECT (D), SE_COMPARATOR_SHOULD_BE_SERIALIZABLE (D), and BC_IMPOSSIBLE_INSTANCEOF (D). Besides, the top four bug types in textbook are not found in the top 20 bug types of OSS.

As a future work, we plan to compare the code level analysis between textbook code examples and Open-Source Software and propose compliant code examples that should be included into textbooks based on the popularity and bug occurrence rate in the OSS.

References

1. E. Lahtinen, K. Ala-Mutka, H. Järvinen. A study of the difficulties of novice programmers, in *Proceedings of the 10th Annual SIGCSE Conference on Innovation and Technology in Computer Science Education* (2005), p. 14–18
2. P. Reimann, T.J. Schult, Turning examples into cases: Acquiring knowledge structures for analogical problem solving. *Educ. Psychol.* **31**(2), 123–132 (1996)
3. J. Mason, D. Pimm, Generic examples: Seeing the general in the particular. *Educ. Stud. Math.* **15**(3), 277–289 (1984)
4. M. Nordström, J. Börstler. Improving OO example programs. Submitted to *IEEE Transactions on Education* (2011), vol. 54
5. J. Sajaniemi, M. Kuittinen, From procedures to objects: A research agenda for the psychology of object-oriented programming education. *Hum. Technol.* **4**(1), 75–91 (2008)
6. D.B. Bois, S. Demeyer, J. Verelst, T.M.M. Temmerman, Does god class decomposition affect comprehensibility? ed. by P. Kokol, in *SE 2006 International Multi-Conference on Software Engineering* (2006), pp. 346–355. IASTED
7. M.E. Caspersen, Educating Novices in The Skills of Programming. PhD thesis, University of Aarhus, Denmark, 2007
8. R. Westfall, ‘hello, world’ considered harmful. *Commun. ACM* **44**(10), 129–130 (2001)
9. CACM, Hello, world gets mixed greetings. *Commun. ACM* **45**(2), 11–15 (2002)
10. M.H. Dodani, Hello world! goodbye skills! *J. Object Technol.* **2**(1), 23–28 (2003)
11. CACM Forum, For programmers, objects are not the only tools. *Commun. ACM* **48**(4), 11–12 (2005)

12. J. Börstler, H.B. Christensen, J. Bennedsen, M. Nordström, L. Kallin Westin, J.-E. Moström, M.E. Caspersen, Evaluating OO example programs for CS1, in *ITiCSE '08: Proceedings of the 13th Annual Conference on Innovation and Technology in Computer Science Education*, (ACM, New York), pp. 47–52
13. M. Nordström, J. Börstler, Heuristics for designing object-oriented examples for novices. Submitted to *ACM transactions on computing education (TOCE)* (2010)
14. FindBugs™ - Find bugs in Java programs (2018). Retrieved from <http://findbugs.sourceforge.net>
15. T. Gaddis, *Starting out with Java: From Control Structures through Objects*, 7th edn. (Pearson Education, New York, 2019)
16. Y.D. Liang, *Introduction to Java Programming and Data Structures, Comprehensive Version*, 11th edn. (Pearson Education, New York, 2019)
17. K. Liu, D. Kim, T.F. Bissyandé, S. Yoo, Y. Le Traon, Mining fix patterns for findbugs violations. *IEEE Trans. Softw. Eng.* **47**(1), 165–188 (2018)
18. FindBugs bug descriptions, <http://findbugs.sourceforge.net/bugDescriptions.html>
19. SpotBugs: Standard bug patterns and bug description, <https://spotbugs.readthedocs.io/en/stable/bugDescriptions.html>

Implications of Blockchain Technology in the Health Domain



Merve Vildan Baysal, Özden Özcan-Top, and Aysu Betin Can

1 Introduction

Blockchains, which are tamper-resistant and tamper-evident distributed digital ledgers, provide a trusted data management system [1]. They are widely known as the technology behind cryptocurrencies such as Bitcoin and Ethereum. However, blockchain has a much wider application area due its benefits such as enabling transparency and traceability of data transactions and providing a secure environment resilient to manipulations. Examples of these application areas include supply chain management, music royalties tracking, personal identity security, and digital voting.

Our research indicated that blockchain applications are widely developed in the health domain as well. There are several commercialized health domain projects which integrate blockchain to their solutions. *Dentacoin* [2] establishes a connection between dental clinics and patients using a blockchain network for patients to receive services from the clinics. *MediBloc* [3] uses blockchain technology for healthcare providers, patients, and researchers to process and manage health data. *SRCoin* [4] is a health information platform that uses blockchain technology to provide safer data storage and processing for healthcare solutions.

Software applications in the health domain are safety-critical; hence, they must be audited by regulatory bodies, such as the Food and Drug Administration (FDA) [5] in the USA and the Medical Device Directives [6] in the EU, to ensure that software applications conform to the regulatory standards. Recently, the FDA announced a *Technology Modernization Action Plan* [7] stating that blockchain technology is in their radar to solve health domain-related challenges. Furthermore, IBM, Walmart, and Merck have been chosen for an FDA pilot program that

M. V. Baysal (✉) · Ö. Özcan-Top · A. B. Can

Information Systems, Informatics Institute, Middle East Technical University, Ankara, Turkey
e-mail: vildan.baysal@metu.edu.tr; ozdenoz@metu.edu.tr; betincan@metu.edu.tr

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_45

641

will explore improving security of medicine supply chains by using blockchain technology [8]. These recent advancements show that the regulatory bodies will continue directing their attention in blockchain studies in the health domain.

While the health domain software is highly regulated, no established standard or guideline exists for blockchain technology employed in this area. The gap regarding essential processes to perform for safe and secure blockchain-based health software needs to be filled. We initiated our studies in this field by examining the existing literature review studies [9–12] conducted on use of blockchain in the health domain. We found that the software development challenges experienced by practitioners in developing blockchain-based health applications and associated solution suggestions were not covered in detail but discussed briefly in these studies. The main focuses of these previous studies were not the practitioners' experiences.

In this study, we present the results of the systematic literature review (SLR) performed to answer the following research questions:

- *RQ1*: What are the application areas of blockchain in the health domain and what is the motivation behind adopting blockchain?
- *RQ2*: What are the challenges of developing health software?
- *RQ3*: To what extent blockchain technology contributes to resolve existing software development challenges in the health domain?
- *RQ4*: Does blockchain introduce new challenges to software development in the health domain?
- *RQ5*: What are existing solution suggestions to the blockchain-related challenges in the health domain?

In this SLR, we focused on practitioners' (such as developers, designers, quality engineers) point of view. Therefore, the SLR includes 27 publications which specifically discuss these research questions from experiences perspective.

2 Background

Blockchain Technology Overview The National Institute of Standards and Technology defines blockchain as tamper-resistant and tamper-evident digital ledgers implemented in a distributed fashion usually without a central authority [1]. The data to be added into a blockchain need multiple approvals before being added in the chain, and it cannot be deleted or changed without approval of network participants once added.

Blockchain networks consist of two types of records: *blocks* and *transactions*. *Transactions* represent the interaction between participants. *Nodes* provide the required computing power to maintain the blockchain in the network. Each node keeps an exact copy of the blockchain [1]. *Blocks* contain transactions and they are formed by the nodes. Each block contains a reference to the previous block. Blocks contain their own hash, data, and the hash of the previous block. This hash is a

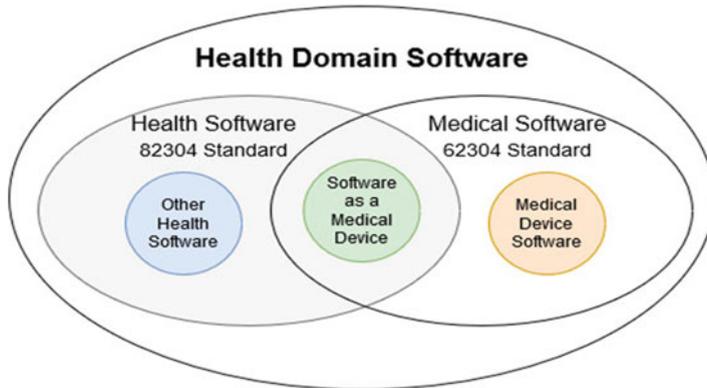


Fig. 1 Health domain software categories. (Adopted from [16])

unique value generated from an input text by using a cryptographic hash function. If anyone changes a single data in the block, the hash of that block changes. Thus, the attempts to change data in blocks can be recognized [1]. In addition to hash algorithms, *consensus models*, which enable a group of mutually distrusting users to work together, are used to secure the blockchain networks [1].

Blockchain technology uses asymmetric-key cryptography to establish a trust relationship between the participants in the network. Asymmetric-key cryptography provides a mechanism to verify the integrity and authenticity of transactions [1].

A *smart contract* is a piece of code stored in the blockchain network [1]. The contract terms defined between the parties are stored with this code, and if a set of predefined terms are met, the smart contract executes itself and the results of the execution are stored on the blockchain.

Health Domain Software Overview Software developed in the health domain has to comply with the associated regulatory requirements. Figure 1, which is based on the study [13], categorizes health domain software with respect to the regulatory compliance requirements.

As shown in Fig. 1, health domain software is categorized into two areas: health software and medical Software. Medical software includes software that is embedded in medical devices and software applications that serve as a medical device (SaMD). Health software both includes SaMDs and other health software.

Medical device software (MDS) is intended to process, analyze, or create medical information [14]. Examples of MDS include software in many devices such as EKG monitor devices, insulin pumps, and medical imaging devices. Software as a medical device (SaMD) is a software used for medical purposes without being part of a hardware [15]. SaMD runs on desktop computers, tablets, smartphones, and smart watches and assists patients and healthcare specialists in treatment planning, medical image viewing, heart rate monitoring, and drug dosage calculating. Other

health software are stand-alone health applications executed on hardware [13]. Examples include healthcare data management and clinical software applications.

As shown in Fig. 1, health software requires compliance with the ISO/IEC 82304 [16] standard, and medical software requires compliance with the IEC 62304 [17] standard. ISO/IEC 82304:2016 health software standard includes processes and practices for guiding healthcare software development. It applies to safety and security of health software designed to operate on general-purpose IT platforms, including mobile devices. IEC 62304:2006 defines requirements that need to be followed in medical device software development.

In addition to these two primary standards, ISO 14971:2009 [18], IEC 80002-1:2009 [19], and IEC 80002-3:2014 [20] are also applied in the health domain. ISO 14971:2009 is an international risk management standard for the manufacture of medical devices. IEC 80002-1:2009 guides the application of ISO 14971 standard to medical device software. IEC 80002-3:2014 is a process reference model that defines software life cycle processes for medical devices driven from IEC 62304:2006.

Software applications in the health domain are audited by regulatory bodies, such as the Food and Drug Administration (FDA) [5] in the USA and the Medical Device Directives [6] in the EU. For instance, the FDA provides several guidance documents for clinical decision support software [21], software as a medical device [22], wireless medical devices [23], and medical device software [24].

3 Research Methodology

A systematic literature review (SLR) is a secondary study which uses a well-defined methodology to answer research questions [25]. There are various guidelines for reviewing the literature systematically. We followed the SLR guideline developed by Kitchenham and Charter [25]. Figure 2 illustrates the overview of our SLR process.

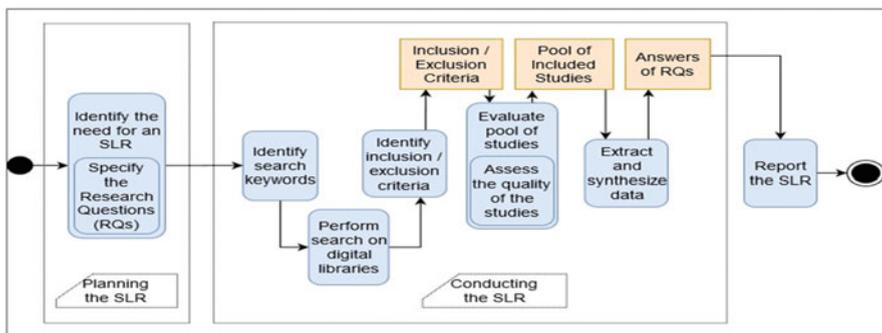


Fig. 2 Phases and steps of our SLR process

3.1 Planning the SLR

The need for performing this review was specified by exploring the existing literature review studies. At this stage, we also formulated the research questions given in Sect. 1. Below, we briefly discuss to what extent these research questions were covered in the previous SLR studies performed on developing blockchain-based applications in the health domain. We did not include studies that do not follow a systematic approach in performing the literature review.

In Table 1, we list the previous SLR studies along with their publication dates, the numbers of the papers included in each of these studies, and the years covered. We also added the columns for our research questions stated in Sect. 1 to highlight the similarities and differences of these SLRs with our study.

As Table 1 shows, although the potential application areas of blockchain in the health domain were listed in all of these studies (RQ1), the motivation behind adopting blockchain in the relevant area was missing in all the SLR studies. Two of the SLR studies discussed the inherited challenges of developing health software (RQ2) [9, 10]. These two studies also discussed the contribution of blockchain technology to resolve these inherited challenges. However, technical details and explanations for the solutions are not provided in the studies. Three studies [9–11] addressed the challenges of developing health applications when blockchain is implemented (RQ4). Among these three studies, only Agbo et al. [11] provided information about possible solutions to overcome these blockchain-related challenges (RQ5). As a result, there is no study which fully explores the five research questions which constitute the main focus of our study.

Other differences of our study are that (i) it includes the latest studies in the literature carried out until March 2020 and (ii) it includes primary studies which contain experiences shared by the practitioners on implementation of blockchain in the health domain. Theoretical papers and secondary studies were not included in our study.

Table 1 Literature review studies reviewing blockchain in the health domain

Ref	Publication date	# of papers included	Years covered	RQ1	RQ2	RQ3	RQ4	RQ5
[12]	2018	33	2015–2018	Partially yes	No	No	No	No
[9]	2019	Not given in the paper	2016–2017	Partially yes	Yes	Partially yes	Yes	No
[10]	2019	39	2016–2019	Partially yes	Yes	Partially yes	Yes	No
[11]	2019	65	2016–2018	Partially yes	No	No	Yes	Yes

3.2 Conducting the SLR

Evaluation Process and Selection of the Publications We performed the search on the IEEE Xplore, ACM libraries, and Google Scholar using the *blockchain AND (healthcare OR health OR medical OR medicine)* keywords. The search which was performed on March 1, 2020, returned 3.363 papers. We applied the following inclusion criteria to present a better scoped research: (i) the papers which share experiences of practitioners/researchers; (ii) the papers presenting blockchain solutions in the health domain; (iii) the papers describing encountered challenges in software development in the health domain; (iv) the papers that conform to specified quality criteria given in Table 3; and (v) the papers that are written in English and accessible.

The first evaluation was based on the titles and the abstracts of the papers, and the second evaluation was performed by reading the full papers. As a result, 27 studies were included in the review process. The number of papers found in the online libraries and the results after each evaluation process are given in Table 2.

Quality Assessment We applied the following criteria to assess the quality of the papers in the literature. While creating Table 3, we used Q1, Q2, and Q3 in the Höst and Runeson's quality checklist as is [26] and added Q4 to these.

All three authors of this paper involved in the quality assessment process. We answered the quality assessment questions for the 77 papers that passed the first evaluation according to three-level indicators: (i) Level 0 (zero) when the criterion was addressed very poorly or not at all, (ii) Level 1 (one) when the criterion was partially addressed, and (iii) Level 2 (two) when the publication had successfully

Table 2 The results of evaluation process

Online library	Initial research	First evaluation result	Second evaluation result
Google Scholar	2.400	47	14
IEEE Xplore	443	21	11
ACM Digital Library	520	9	2
Total	3363	77	27

Table 3 Quality assessment questions

ID	Quality assessment query	Quality Indicator (0–2)
Q1	Are the authors' intentions with the research made clear?	0 – No 1 – Partially 2 – Yes
Q2	Does the study contain conclusions and implications for practice and future research?	0 – No 1 – Partially 2 – Yes
Q3	Does the study give a realistic and credible impression?	0 – No 1 – Partially 2 – Yes
Q4	Are the challenges or solutions adequately defined?	0 – No 1 – Partially 2 – Yes

satisfied the criterion. We included the studies in the review with four or higher rating.

Data Extraction We used a spreadsheet which was collectively managed by all of the authors to extract data from the publications and review the data. We collected the bibliometric data of the papers (i.e., publication type, publication year, the number of citation) and all the information needed to address our research questions.

4 Results and Discussion

In this section, we first present the bibliometric overview of the publications, and then we present the answers to the research questions.

The result publication set consists of 27 papers, among which 13 were published in journals and 14 were published in conference proceedings. The publication years of the included papers are plotted in Fig. 3a. Although blockchain was introduced in 2008 by Satoshi Nakamoto, we did not come across any publications on use of blockchain in the health domain before 2016. This graph shows that there is an increase in the number of publications in 2016, 2017, and 2018. Although it may seem that there is a decrease in numbers in the recent years, we think that the interest in this subject would increase.

We present the citation number ranges of the publications in Fig. 3b as indicators of the interest in the field. According to the Google Scholar data, just three of the publications were cited at a low rate (less than ten citations per publication). The reason for low citation numbers might be the publication date of these papers which is 2019 and 2020. The number of citations of all the remaining studies is quite high considering the publication dates start from 2016.

Next, we present a discussion on the data retrieved from the analyzed papers based on the five research questions defined in Sect. 1.

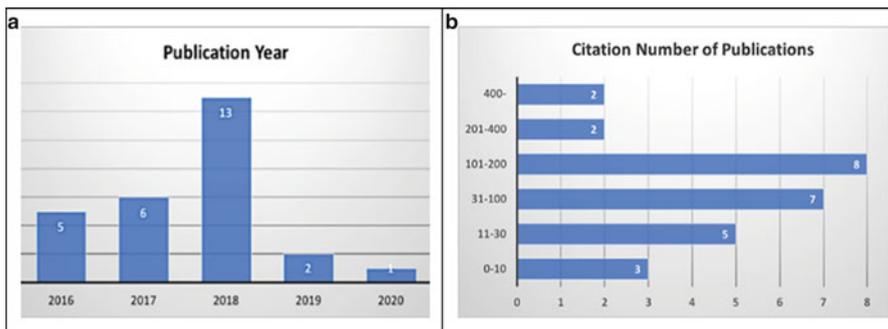


Fig. 3 (a) Publication years of the included papers (b) Citation numbers of the included papers

RQ1. What are the application areas of blockchain in the health domain and what is the motivation behind adopting blockchain in the health domain?

We extracted the main application areas of blockchain in the health domain, the motivation behind adopting this technology in these areas, and examples of blockchain-oriented solutions (see Table 4). The majority of the publications indicate the use of blockchain technology for electronic health and medical record management (13 papers). The second popular application area is the Internet of Medical Things with seven papers. Other application areas are medicine supply chain management (3 papers), clinical trials (2 papers), and precision medicine (2 papers).

RQ2. What are the challenges of developing health software? RQ3. To what extent blockchain technology contributes to resolve these existing software development challenges in the health domain?

We grouped the challenges and associated solution suggestions of the publications under three main headings: (1) meeting regulatory requirements, (2) security and protection of privacy, and (3) ensuring interoperability.

(1) Meeting Regulatory Requirements

Challenge 1.1 Regulatory bodies require monitoring product supply chain before and during the distribution of medicines to prevent distribution of falsified drugs [27].

Solution 1.1 The blockchain technology and smart contract-based structures enable monitoring of product supply chain through detecting data anomalies, unauthorized data insertions, missing raw materials, and identifying authorized drug vendors/manufacturers and storage of medicine information. Blockchain, which validates and authenticates transactions, enables monitoring both the participants of the system and each movement of medicines across the medicine supply chain [27]. The primary structure used in blockchain for this problem is the smart contracts. In this context, an FDA account added to the blockchain network will be notified by a smart contract whenever a transaction is executed in the supply chain, such as production, shipment, and receipt of medicine. By his way, the FDA account could verify each manufacturing pedigree by checking the credentials of the manufacturer [27] at each transaction.

Challenge 1.2 Regulatory bodies require availability of methods and results of all clinical trials; however, more than half of the trials have failed to provide this data to regulatory bodies [30]. In addition, a recent study highlights that there is a high risk of data manipulation in clinical trials [51]. Data that might be subject to manipulation include subject registration, trial registration, and clinical measurements [30].

Solution 1.2 Due to tamper-resistant characteristics, blockchains could be used to prevent data manipulation in clinical trials [30]. Smart contracts are used to manage the clinical trial life cycle including trial registration, regulatory approval, recruiting study subjects, and data entry processes. This way, the clinical trial results could be

Table 4 Blockchain in the health domain

Application areas	Motivation behind adopting blockchain in the relevant area	Examples of blockchain-oriented solutions
Medicine supply chain management	Difficulty of identifying unauthorized medicines Difficulty of specifying falsified medicines that misrepresent their content or source	Sylim et al. [27] developed a pharmacosurveillance blockchain system. Tseng et al. [28] developed the Gcoin blockchain application for governance of whole medicine supply chain life cycle. Lu and Xu [29] developed the originChain blockchain application to ensure medicine data's availability to service providers and to automate regulatory-compliance checking in medicine supply chain
Clinical trials	Risk of clinical trial data manipulation The need for providing data transparency in clinical trials for scientific reliability of the findings The need for sharing and ensuring traceability of clinical trial data The need for structuring clinical trial data which is usually kept in silo forms	Nugent et al. [30] developed smart contracts, which are pieces of code that can be executed automatically based on predefined conditional triggers, on a private Ethereum network to enable data transparency, prevent data manipulation, and ensure scientific reliability in clinical trials. Shae et al. [31] developed a four-layered system architecture for development of blockchain-based applications for clinical trials, precision medicine, and assisting medical decision-making process
Precision medicine	The need for ensuring privacy and security of data in diagnosing, treating, and preventing diseases by considering the variabilities in genes, environment, and lifestyle of individuals	Juneja and Marefat's [32] system uses deep learning for arrhythmia classification and smart contracts for keeping access control policies. Lee and Yang [33] developed a nail analysis system that uses microscopy sensors and blockchain together for effective prediction of fingernail diseases' diagnosis
Internet of Medical Things	The need for a secure system in collecting and sharing data in a real time manner via IoT technology (e.g., body scanners, wearable devices, and heart monitors)	Griggs et al. [34] evaluate patients' data collected via IoT healthcare devices based on customized threshold values stored in smart contracts. Saravanan et al. [35] developed a smart contract-based IoT system for diabetic patients. Jita and Pieterse [36] presented an architectural design for homecare system development that uses smart devices for monitoring patient's vitals and blockchain to store data. Liang et al. [37] proposed a user-centric health data sharing solution using a blockchain. Dey et al. [38] and Pham et al. [39] developed two different blockchain-based IoT solutions, both of which use biosensors to measure medical condition of patients and store it in blockchain. Uddin et al. [40] proposed an architecture for development of a continuous patient monitoring system. In this architecture, an agent, called patient-centric agent, manages a blockchain component to preserve privacy when data streaming from body area sensors need to be stored securely
Electronic health/medical record management	The need for systems to be secure against attacks due to the sensitivity of patient data in electronic health records (EHR) The need for patient data to be up-to-date and available when needed	[41–46] provide EHR solutions for effective and secure storage of patients' medical data using blockchain. Due to their decentralized structure and cryptographic functions, blockchains prevents hackers to breach or corrupt the data, and data is kept up-to-date [47–50, 53–55] propose blockchain-based EHR sharing solutions. In these solutions, the accountability and transparency of transactions are maintained during data-sharing process by using blockchain technology

reported to the regulatory agencies during the clinical trial execution. In addition, smart contracts could be used to increase transparency in reporting clinical trial data by capturing the data that might be intended to be manipulated [28, 30].

Challenge 1.3 The Health Insurance Portability and Accountability Act (HIPAA) provides standards for healthcare-related electronic transactions [52]. The Privacy Rule of HIPAA requires protecting the privacy of health information. Therefore, patient data should be stored in an anonymous manner [32, 41].

Solution 1.3 Blockchain enables pseudo-anonymity, which means users are anonymous, but their account identifiers are not [34]. This pseudo-anonymity structure gives patients' the option to hide their identities with alphanumeric addresses and prove their identity to others when needed [36, 37, 41].

(2) *Security and Protection of Privacy*

Challenge 2.1 Secure and reliable storage and transmission for medical data need to be ensured [36, 38, 41, 43, 44, 47, 48]. Since health data has high economic value, it is a lucrative target for criminals [34]. Criminals could attack the healthcare system and abuse personal health information of the patients. Especially, IoT devices used for remote patient monitoring are vulnerable to cyberattacks and data theft [39].

Solution 2.1.a Centralized systems present a single point of failure by nature which means if the center component fails or is compromised, the whole system stops working. On the other hand, blockchain has a decentralized structure which makes the system robust and resilient to cyberattacks [38, 40, 41].

Solution 2.1.b Each block in a chain can be used to keep permanent logs of every data transmission [32, 36, 41, 44, 48] such as data retrieval requests and updates from health service providers. This transparency provides data theft resistance. In addition, audit trails in blockchain allow tracing who made what changes when [45].

Solution 2.1.c Smart contracts can be a solution for privacy and security issues of personal health information by logging patient-provider relationships and permissions [44] and generating unchanged logs on transactions [39].

Solution 2.1.d All data and transactions in blockchain are encrypted. Thus, unauthorized access or theft of health data can be avoided [35, 38, 41, 43]. Blockchain uses asymmetric cryptography to protect data on the network and to authenticate users. This allows secure data sharing between participants of a system [35].

Challenge 2.2 As health data is very sensitive, patients of health systems should have control over their data [40, 41, 43, 49, 53]. Authorization of with whom the health data will be shared should be in the patient, not in third parties and institutions.

Solution 2.2 Blockchain provides data storage and sharing without violating patient rights by giving patients the control of their own health data. As patients' health data is stored in the blockchain using asymmetric encryption algorithm, the

patient retains control over each transaction to access her/his health data. Only trusted parties that the patient gives access permission reaches to the data [36, 39–41, 43, 46, 54].

(3) *Ensuring Interoperability*

Challenge 3.1 Procedures to regulate data transfers between healthcare providers are not well defined. There are deficiencies in coordinated data sharing. Interoperability issues between different healthcare providers create additional barriers to effective data sharing [43, 53].

Solution 3.1 As patients would have access to their own health data in a blockchain, they can transfer their health data to the healthcare service providers when needed instead of service providers' transferring the data to each other. Thus, it is possible to avoid undefined procedures and interoperability issues in sharing data among healthcare providers [45].

Challenge 3.2 Mobility of patients requires cross-border exchange of patient data. This fact creates the challenge of dealing with different privacy and data protection requirements of different countries [50].

Solution 3.2 Different data share policies need to be considered in designing blockchains and smart contracts [49, 50]. We also suggest to design a structure which allows each network participating country to implement specific policies for the protection and control of health-related data.

RQ4. Does blockchain introduce new challenges to software development in the health domain? RQ5. What are existing solution suggestions to the blockchain-related challenges in the health domain?

We present blockchain-related challenges along with solution suggestions below.

Challenge 1 In blockchain, data cannot be altered or deleted after it is stored in a blockchain. However, due to protection laws of health data, it is necessary to erase it when a user requests to do so [49].

Solution 1 By storing health data in an external storage and its hash in the blockchain, it would be possible to avoid this challenge [32, 44, 49]. In this way, the health data in the external storage can be deleted if it needs to be deleted. The hash value that is not connected to the data will continue to exist in the blockchain.

Challenge 2 As health data may consist of images, and treatment plans, the size of the data can be large [54]. This might cause storage issues [29].

Solution 2 Storing large data in an external storage and keeping the hash of the data in the blockchain would be solution to this problem as well [32, 44, 49] without compromising the tamper-resistant nature of the blockchain. The hash of the large data is embedded inside a digitally signed transaction, which was included in the blockchain by consensus. The origin and timestamp of the data in the external storage can be confirmed when the hash for that data matches a hash in

the blockchain. When the data in the external storage changes, the hash of the data also changes; thus, data manipulations could be detected.

Challenge 3 Blockchain introduces performance challenges [39, 43, 54]. It is not possible to run executions in parallel in blockchain, since each node repeats the same process for mining the subsequent block. This affects the efficiency of the system and may create bandwidth and response time problems.

Solution 3 A solution that overcomes the performance challenge has not yet been proposed by the papers in our pool. However, the choice of architectural design may have a positive impact on the performance: In a blockchain network, we have to decide on a consensus model, which has an effect on performance while assuring the validity of transactions. Another design decision is to have public or private blockchain [55]. If high performance is essential for an application, private blockchain networks with trusted nodes may be preferred.

Challenge 4 Not all individuals are capable of handling their medical data themselves in blockchain such as giving consent for data use. Additionally, data providers may not be in a culture to release the control of the data [45].

Solution 4 Solutions to overcome these challenges were not proposed yet.

Challenge 5 Blockchain development introduces specific constraints to the development processes.

Challenge 5.1 Once a smart contract is deployed to the blockchain, it cannot be modified or replaced. A new contract has to be created when a change request is received [56, 57].

Solution 5.1 Although this challenge reminds us running a waterfall-like plan-driven development process for smart contract development, it would not guarantee developing error-free features. This problem may be solved by developing new software design principles to contribute in the development of high-quality smart contract.

Challenge 5.2 As part of the development process, smart contracts need to be tested in production environments. The cost of testing a smart contract in the production environment includes an execution fee (called *gas price* in Ethereum). This cost varies based on the operations in the smart contract. Calculating the cost required for executing a smart contract on a blockchain network is not easy, especially for large-scale projects where smart contracts have complex coding [42].

Solution 5.2 A solution to this challenge has not yet been proposed. However, in test networks, smart contract testing is performed without price. To reduce the cost, detailed test processes are needed to be performed in the test network before uploading smart contracts to the production environment.

Challenge 5.3 In blockchain-based application development at the unit testing phase, many of the units highly depend on the collective inputs of other units. Therefore, not all of the units could be tested individually. At the acceptance testing

phase, collective inputs of the participants in the blockchain network are required. Thus, unit and acceptance testing phases require complete implementation of the system [42].

Solution 5.3 This problem may be solved by developing a Software Development Life Cycle (SDLC) to meet specifically the testing requirements of blockchain-based applications.

5 Conclusion and Future Work

In this study, we performed a systematic literature review to uncover the application areas of blockchain in the health domain, the motivation behind adopting blockchain, the challenges of developing health software, blockchain's contribution to resolve the existing software development challenges, the challenges introduced by blockchain to health software development, and existing solution suggestions to the blockchain-related challenges. In addition to the suggestions of the studies in our paper pool, we also included our own solution suggestions to these challenges.

We found that the main application areas of blockchain in the health domain are electronic health and medical record management, Internet of Medical Things, medicine supply chain management, clinical trials, and precision medicine areas.

Blockchain has a significant potential in contributing to the inherent health domain challenges such as ensuring regulatory requirements and dealing with security, privacy, and interoperability issues. Blockchain contributes to resolve these challenges by enabling regulatory bodies to monitor supply chains, preventing data manipulation and data theft, increasing transparency in transactions, and giving patients the control of their own health data. However, most of the solutions that address the health domain-related challenges are at the proof of concept level.

While providing solutions to the inherent health domain challenges, blockchain introduces new challenges which are data protection, data size, performance, individual's data management, and development process issues. We summarize solution suggestions to these challenges as follows: addressing data protection and size problems by storing health data in an external storage and its hash in the blockchain, increasing performance with architectural design decisions, and addressing development process issues by developing new software design principles and by developing a SDLC to meet specific requirements of blockchain-based applications.

Blockchain technology introduces specific constraints to development processes in the health domain. Structural differences of blockchain such as the immutability feature, collective collaboration of network participants, and differences in testing processes cause current SDLC models not be fully compatible with the development process of blockchain-based applications. To meet the requirements of the development of blockchain-based applications, it is necessary to develop a new model specific to the development of these applications.

References

1. D. Yaga, P. Mell, N. Roby, K. Scarfone, Blockchain technology overview - NIST (2018)
2. Dentacoin ecosystem, (2020), <https://dentacoin.com>. Accessed 10 Apr 2020
3. MediBloc, (2020), <https://medibloc.org/en>. Accessed 11 Apr 2020
4. SRcoin, (2020), <https://www.srcoin.info>. Accessed 11 Apr 2020
5. The FDA, <https://www.fda.gov/home>. Accessed 11 Dec 2019
6. European Commission, https://ec.europa.eu/growth/sectors/medical-devices/current-directives_en. Accessed 10 Dec 2019
7. FDA, FDA's Technology Modernization Action Plan (TMAP), (2019), <https://www.fda.gov/media/130883/download>. Accessed 10 Dec 2019
8. Reuters, IBM, Walmart, Merck in blockchain collaboration with FDA, <https://www.reuters.com/article/us-fda-blockchain/ibm-walmart-merck-in-blockchain-collaboration-with-fda-idUSKCN1TE1SA>. Accessed 10 Dec 2019
9. T. McGhin, K.R. Choo, C.Z. Liu, Blockchain in healthcare applications: Research challenges and opportunities. *J. Netw. Comput. Appl.* **135**(1), 62–75 (2019)
10. S. Yaqoob, M. Khan, R. Talib, A. Butt, S. Saleem, F. Arif, A. Nadeem, Use of blockchain in healthcare: A systematic literature review. *Int. J. Adv. Comput. Sci. Appl.* **10**(5), 644–653 (2019)
11. C. Agbo, Q. Mahmoud, J. Eklund, Blockchain technology in healthcare: A systematic review. *Healthcare* **7**(2), 56 (2019)
12. M. Hölbl, M. Kompara, A. Kamišalić, L.N. Zlatolas, A systematic review of the use of blockchain in healthcare. *Symmetry* **10**(10), 470 (2018)
13. G. Heidenreich, Scope of IEC health software standards, in TOPRA Annual Medical Devices Symposium, (2014)
14. J. Jensen, I. Sandoval-watt, Software in medical devices, https://www.advamed.org/sites/default/files/resource/software_in_medical_devices_-_module_1.pdf. Accessed 05 Dec 2019
15. IMDRF, IMDRF Software as a Medical Device (SaMD), (2013). Available: <http://www.imdrf.org/docs/imdrf/final/technical/imdrf-tech-131209-samd-key-definitions-140901.pdf>. Accessed 05 Dec 2019
16. IEC 82304-1:2016 health software - part 1: General requirements for product safety, (2016). Available: <https://www.iso.org/standard/59543.html>
17. IEC 62304:2006 medical device software - software life cycle processes, (2006). Available: <https://www.iso.org/standard/38421.html>
18. ISO 14971:2019 medical devices - application of risk management to medical devices, (2019). Available: <https://www.iso.org/standard/72704.html>
19. IEC/TR 80002-1:2009 medical device software - part 1, (2009). Available: <https://www.iso.org/standard/54146.html>
20. IEC/TR 80002-3:2014 medical device software - part 3, (2014). Available: <https://www.iso.org/standard/65624.html>
21. U.S. FDA, Clinical decision support software, (2019). Available: <https://www.fda.gov/media/109618/download>. Accessed 06 Dec 2019
22. US FDA, Software as a Medical Device (SaMD): Clinical evaluation, (2017). Available: <https://www.fda.gov/media/100714/download>. Accessed 06 Dec 2019
23. US FDA, Wireless medical devices, (2014). <https://www.fda.gov/medical-devices/digital-health/wireless-medical-devices>. Accessed 06 Dec 2019
24. U.S. FDA, General principles of software validation, (2002). Available: <http://www.fda.gov/downloads/RegulatoryInformation/Guidances/ucm126955.pdf>. Accessed 06 Dec 2019
25. B. Kitchenham, O. Pearl Brereton, D. Budgen, M. Turner, J. Bailey, S. Linkman, Systematic literature reviews in software engineering - a systematic literature review. *Inf. Softw. Technol.* **51**(1), 7–15 (2009)
26. M. Höst, P. Runeson, Checklists for software engineering case study research, in *Int. Symposium on Empirical Software Engineering and Measurement*, (2007), pp. 482–484

27. P. Sylim, F. Liu, A. Marcelo, P. Fontelo, Blockchain technology for detecting falsified and substandard drugs in distribution: Pharmaceutical supply chain intervention. *J. Med. Int. Res. Protoc.* **20**(9), 1–12 (2018)
28. J.H. Tseng, Y.C. Liao, B. Chong, S.W. Liao, Governance on the drug supply chain via gcoin blockchain. *Int. J. Environ. Res. Public Health* **15**(6), 1055 (2018)
29. Q. Lu, X. Xu, Adaptable blockchain- based systems: A case study for product traceability. *IEEE Softw.* **34**(6), 21–27 (2017)
30. T. Nugent, D. Upton, M. Cimpoesu, Improving data transparency in clinical trials using blockchain smart contracts. *F1000Research* **5**, 1–9 (2016)
31. Z. Shae, J.J.P. Tsai, On the design of a blockchain platform for clinical trial and precision medicine, in *Int. Conf. on Distributed Computing Systems*, (2017), pp. 1972–1980
32. A. Juneja and M. Marefat, Leveraging blockchain for retraining deep learning architecture in patient-specific arrhythmia classification,” in *Int. Conf. on Biomedical and Health Informatics*, (2018), pp. 393–397
33. S.H. Lee, C.S. Yang, Fingernail analysis management system using microscopy sensor and blockchain technology. *Int. J. Distributed Sensor Netw.* **14**(3), 155014771876704 (2018)
34. K.N. Griggs, O. Ossipova, C.P. Kohlios, A.N. Baccharini, E.A. Howson, T. Hayajneh, Healthcare blockchain system using smart contracts for secure automated remote patient monitoring. *J. Med. Syst.* **42**(7), 1–7 (2018)
35. M. Saravanan, R. Shubha, A.M. Marks, V. Iyer, SMEAD: A secured mobile enabled assisting device for diabetics monitoring, in *Int. Conf. on Advanced Networks and Telecommunications Systems*, (2018), pp. 1–6
36. H. Jita, V. Pieterse, A framework to apply the internet of things for medical care in a home environment, in *Int. Conf. on Cloud Computing and IoT*, (2018), pp. 45–54
37. X. Liang, J. Zhao, S. Shetty, J. Liu, D. Li, Integrating blockchain for data sharing and collaboration in mobile healthcare applications, in *Int. Symposium on Personal, Indoor and Mobile Radio Communications*, (2018), pp. 1–5
38. T. Dey, S. Jaiswal, S. Sunderkrishnan, N. Katre, HealthSense: A medical use case of IoT and blockchain, in *Int. Conf. on Intelligent Sustainable Syst*, (2018), pp. 486–491
39. H.L. Pham, T.H. Tran, Y. Nakashima, A secure remote healthcare system for hospital using blockchain smart contract, in *Globecom Workshops*, (2019), pp. 1–6
40. M.A. Uddin, A. Stranieri, I. Gondal, V. Balasubramanian, Continuous patient monitoring with a patient centric agent: A block architecture. *IEEE Access* **6**, 32700–32726 (2018)
41. D. Ivan, Moving toward a blockchain-based method for the secure storage of patient records, in *NIST Workshop on Blockchain & Healthcare*, (2016), p. 11
42. M.H. Miraz, M. Ali, Blockchain enabled smart contract based applications: Deficiencies with the SDLC models. *Baltica* **33**(1), 101–116 (2020)
43. A. Ekblaw, A. Azaria, J.D. Halamka, A. Lippman, A case study for blockchain in healthcare. *Open & Big Data Conference* **13**, 13 (2016)
44. A. Azaria, A. Ekblaw, T. Vieira, A. Lippman, MedRec: Using blockchain for medical data access and permission management, in *Int. Conf. on Open and Big Data*, (2016), pp. 25–30
45. N. Kshetri, Blockchain and electronic healthcare records. *IEEE Computer* **51**(12), 59–63 (2018)
46. M.T. de Oliveira, et al., Towards a blockchain-based secure electronic medical record for healthcare applications, in *Int. Conf. on Communications*, (2019)
47. Q. Xia, E.B. Sifah, A. Smahi, S. Amofa, X. Zhang, BBDS: Blockchain-based data sharing for electronic medical records in cloud environments. *Information* **8**(2), 44 (2017)
48. Q.I. Xia, E.B. Sifah, K.O. Asamoah, J. Gao, X. Du, M. Guizani, MeDShare : Trust-less medical data sharing among. *IEEE Access* **5**, 1–10 (2017)
49. C. Esposito, A. De Santis, G. Tortora, H. Chang, K.K.R. Choo, Blockchain: A panacea for healthcare cloud-based data security and privacy? *Cloud Computing* **5**(1), 31–37 (2018)
50. L. Castaldo, V. Cinque, Blockchain-based logging for the cross-border exchange of eHealth data in Europe, in *Security in Computer and Information Sciences*, (2018)

51. COMPARE tracking switched outcomes in clinical trials. Available: <https://compare-trials.org/results>. Accessed 02 Jan 2020
52. HIPAA, <https://www.hipaajournal.com/hipaa-privacy-laws/>. Accessed 02 Jan 2020
53. K. Fan, S. Wang, Y. Ren, H. Li, Y. Yang, MedBlock: Efficient and secure medical data sharing via blockchain. *J. Med. Syst.* **42**(8), 1–11 (2018)
54. A. Dubovitskaya, Z. Xu, S. Ryu, M. Schumacher, F. Wang, Secure and trustable electronic medical records sharing using blockchain, in *AMIA Symposium*, (2017), pp. 650–659
55. X. Xu, et al., A taxonomy of blockchain-based systems for architecture design, in *Int. Conf. on Software Architecture*, pp. 243–252, (2017)
56. C. Sillaber, B. Wailt, Life cycle of smart contracts in blockchain ecosystems. *Datenschutz und Datensicherheit* **41**(8), 497–500 (2017)
57. R. Koul, Blockchain oriented software testing - challenges and approaches, in *Int. Conf. for Convergence in Technology*, pp. 1–6, (2018)

A Framework for Developing Custom Live Streaming Multimedia Apps



Abdul-Rahman Mawlood-Yunis

1 Introduction

The rise in the number of mobile-connected devices and the emergence of new streaming models, such as on-demand, on-the-go, and interactive streaming, has an impact on the viewing practice of consumers. Mobile devices will play a major role in new viewing habit changes [7, 9]. In this paper, we present a generic framework for developing custom live streaming multimedia apps for mobile devices, i.e., we identify the principal characteristics of mobile live streaming apps, the components, components' interactions, and the design decision needed to create such apps. The framework can help developers in different ways. It can be used as a starting point for designing live streaming app architecture, identifying the list of components needed to develop custom live streaming apps, distributing work among the development team, assessing the usability of individual components, and creating domain vocabularies enabling discussions and understudying among developers.

To demonstrate how the generic framework can be used, we build an app for live streaming audio for Android devices using URLs. The app acts as an instance of the framework and validates it. It is also beneficial to both developers and end users. Developers can reuse the instance app structure to create and publish new apps, and end users can customize the app to their own specific needs using live streaming URLs. For example, users can group their favorite music and news stations in one place and easily play and switch from one station to another. The app turns the user's device into a radio and lets them listen to live streaming stations while in the office, on the road, or in any other setting. The app is like Spotify but on a smaller scale

A.-R. Mawlood-Yunis (✉)

Physics and Computer Science Department, Wilfrid Laurier University, Waterloo, Canada
e-mail: amawloodyunis@wlu.ca

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_46

657

(that's probably all you need; your favorite station and not all the stations that come with Spotify).

The paper has two main contributions: (1) the generic framework which can be adapted when developing live streaming multimedia apps or similar ones and (2) using new URLs, customizable live streaming apps can be created. The app's source code and complete documentation can be used by instructors to teach various mobile topics. The source code and complete documentation are available upon request.

This paper is organized as follows: in Sect. 2, we describe the generic framework components and their functionalities; in Sect. 3, we present the framework architecture and its uses; in Sect. 4, we describe an instance of the framework, the functionalities, and the trade-offs using different components and approaches to create an instance app; and in Sect. 5, we conclude the paper and describe some future works.

2 A Framework for Multimedia Live Streaming Apps

The framework is made of 11 key constructs or components. The components are user interface, background process or service, communication channel, media player, power management, wake lock, thread, file management, URL, data storage, and network permissions. The ten components comprise the minimum components required for any live streaming framework. In the following, we describe each of these model components, i.e., we discuss the functionality and the trade-offs using different components and approaches. We also present the class diagram, and the architecture, of the framework.

2.1 App Interface

The app interface is one of the main components of the framework. It is required for (1) interacting with the user; (2) communicating with the background process, for example, to start and stop the content player component running in the background; and (3) listening to the message broadcast from the background process.

2.2 Background Process

A background process is needed to perform long-running tasks, i.e., playback streams in the background with no graphical user interface. Once the process starts, it might continue running even if the original application is ended or the user moves to a different application. The run continuity is platform and setup dependent. The background process is the right choice to use when the process does not interact

with the user, i.e., is not the forefront process. The process can be defined as private or public. When private, it is usable only by the app it belongs to; however, when public, it is usable by other apps, i.e., other apps can start process using process API (application programming interfaces).

2.3 *Communication Channel*

A communication channel component is needed to receive and handle broadcast messages sent back and forth between foreground and background processes via method calls. The communication channel component does not need to have a user interface, but it can create notifications to alert the user when a broadcast event occurs. The communication channel object needs to be instantiated and registered to process the broadcasted messages on arrival.

2.4 *Content Player*

A media content player component, such as Android Media Player, VideoView, or ExoPlayer, is needed to control the playback of multimedia streams. The media player needs to be prepared, started, and ultimately released. There are two ways for the player to enter the prepared state: *synchronous* or *asynchronous* way. The difference between the two is in what thread they are executed. The *synchronous* one executes in the foreground thread or the user interface thread, and the *asynchronous* one is executed in the background thread. To avoid users getting an ANR (application not responding) message, the asynchronous way needs to be used for playing the live data over stream. Additional actions might need to be taken in case the media player content is not prepared instantly. For example, a listener object can be set for informing when the media player can start.

2.5 *Power Manager*

If mobile devices go into a low-power state, it will prevent apps from running. To control the power state on devices, you need to use power management to keep the CPU running, prevent screen dimming or going off, or prevent the backlight from turning on.

2.6 Network Lock

Acquiring the Wi-Fi lock keeps the connection on until the application releases the lock. Live streaming apps need to keep the Wi-Fi component awake; hence, you need to acquire the lock. A design decision needs to be made whether to keep the app running even when the device screen is off.

2.7 Threads

Preparing media content asynchronously is not enough to avoid ANR prompts and your application hanging when playing live streaming multimedia apps. The best solution is to run the media content player instance in its thread, i.e., run Media Player in a separate thread within the service. The code snippet below shows one way how this step can be achieved when Android is used.

```
@Override
onPrepared(MediaPlayer mp) {
new Thread(new Runnable() {
    volatile boolean running = true;
    public void run() {
        try {
            if (null != MainActivity.url) {
                if (!player.isPlaying()) {
player.start();

                    bufferingComplete();
                }
            }
        } catch (Exception e) {
            player.reset();
        }
    }
}).start();
}
```

2.8 Data Storage

Storing data and operations such as reading, writing, updating, viewing, and deleting data are needed for almost all kinds of apps. Data can be saved in the database, in local files, or in the cloud and in different formats. For live streaming multimedia-type apps, the data most likely is limited to network URLs and icons of stream providers, user preferences, and settings.

2.9 URLs

When apps come with built-in streaming services such as a preset list of radio stations, TV channels, and live streaming events, the stream provider names the URLs and icons needed to be saved or referenced at the URL component. This enables easy app extensions and changes. For example, if you want to add a new station to your preset list, then you simply add the new station to the list of existing stations. There is no need to change other parts of the code.

2.10 User Permissions

For an app to read, write, update, view, or delete data, it must have permission from the user to do so. Similarly, for an app to have access to Wi-Fi and network, it needs the user's permission. The request for these permissions needs to be included in the app's code for the app to operate.

2.11 Other Components

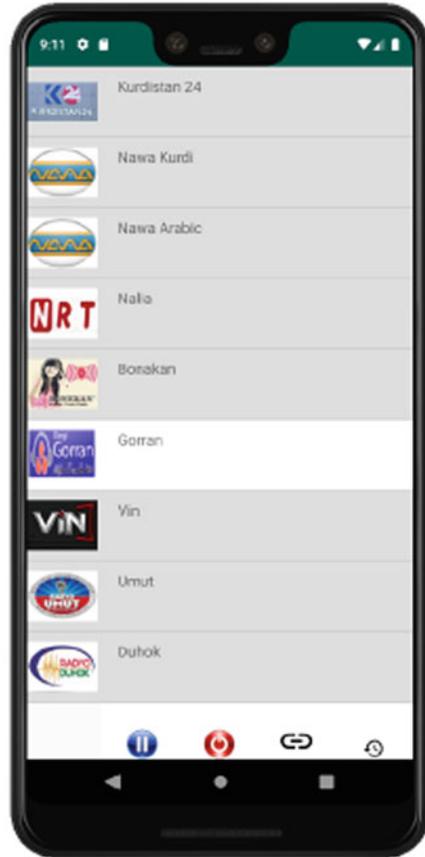
There are other components live streaming apps might require. For example, apps might need to handle which app has *audio focus* when more than one app plays audio to the same output device. Here, the audio focus is not considered. This is because some systems, e.g., Android, have a built-in solution for the audio focus [1], meaning, when a second app requests audio, the currently running one pauses playing or lowers its volume to give the second one audio focus.

3 Framework Class Structure

In Fig. 1, all the framework components are put together, and the class diagram, i.e., the framework architecture, has been created. The class diagram shows the components and the dependency between model components. When not identified, the cardinality relationship between components is one-to-one, and the relationship type is an association.

Fig. 2 App interface

IV. FRAMEWORK IMPLEMENTATION



In implementing the framework, we use *Android Service* and *MediaPlayer* components to play live streaming radio stations using URLs. We also use *BroadcastReceiver* as a communication channel between the service running in the background and the app interface, i.e., the *MainActivity*. Users start the app from the main screen which gets the radio URL from the *URLList* class and starts the *MediaPlayer*. The current app is a refactored and extended version of our previous work presented at [1]. The new features are described in Sect. 4.1.

Below, we describe the app components, implementations, and steps needed to develop such apps. We discuss the functionality and the trade-offs using different components and approaches. We also present the class structure, i.e., the architecture, of the app. The app interface is presented in Fig. 2.

A. *App new features*

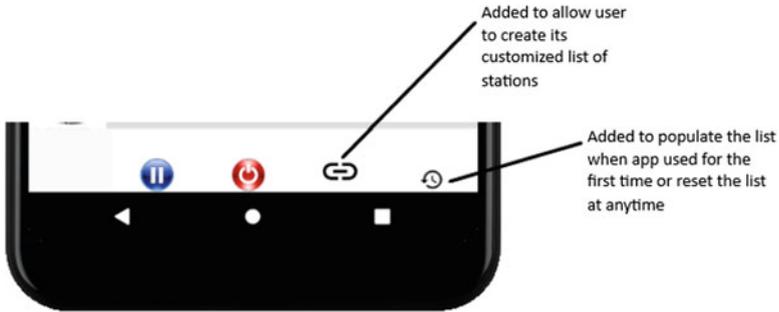


Fig. 3 App new features highlighted

4.1 *App New Features*

To validate the framework, we have refactored and extended our live streaming app presented in [1]. In this section, we elaborate shortly on the three new features added to the current version of the app. First, in the current version, users can delete any listed radio station. This is done by pressing and holding on an item in the list for a short period and confirming the delete by pressing the ok button on the popup dialog box. Second, users can add new stations to the list. This is done by pressing on the *link button* at the bottom of the screen and following the wizard which prompts users to enter the station name, link, and icon in sequence. This feature enables users to create a customized list of stations. Lastly, users can populate the list with predefined embedded stations when using the app for the first time or reset the list at any time by pressing the *reset* button. Figure 3 shows both the link and reset buttons at the bottom of the app.

4.2 *Main Activity*

The app's user interface and background service setup are all done at the MainActivity class. MainActivity can communicate with the background service to start and stop the MediaPlayer and listens to the message broadcast from the service, i.e., it maintains a reference to the service, makes calls on the service just as any other class, and can directly access members and methods of the service.

4.3 Service

Service [5, 6] is an app component that performs long-running tasks in the background with no graphical user interface. Once service starts, it can continue to run even if the original application is ended, or the user moves to a different app. Whether service runs continuously is platform and service setup dependent. We run service indefinitely until explicitly stopped and restart it if the Android system terminates it for any reason. Service is the right choice to use when an activity does not interact with the user, i.e., is not the forefront activity. Service can be private or public. When private, it is usable only by the app it belongs to; however, when it is public, it is usable by apps other than the app it belongs to, i.e., another app component can start Service using a call to the API. In the current app, the MainActivity component starts service with the method call `startForegroundService()` followed by calling `startforeground(int, Notification)` by the service.

4.4 Message Broadcast Receiver

We use BroadcastReceiver [2] as a communication channel to receive and handle broadcast messages sent from service by the `sendBroadcast(Intent)` method. It is another entry point to the app. The broadcast class does not have a user interface but can create a status bar notification to alert the user when a broadcast event occurs. The Android system delivers a broadcast Intent to all interested (registered) broadcast receivers. Apps can initiate broadcast messages to let other apps know, for example, that some data has been downloaded to the device and is available to use.

The BroadcastReceiver needs to be instantiated and registered to process the broadcasted messages on arrival. The four steps involved in message broadcast and receive are:

- Create the BroadcastReceiver object.
- Register BroadcastReceiver object to receiving messages.
- Message broadcasting.
- Actions performed upon receiving the broadcasted message.

4.5 Media Player

We use the MediaPlayer [3] class to control the playback of radio streams. MediaPlayer needs to be prepared, started, and ultimately released. The MediaPlayer's life cycle shows that the player must first enter the prepared state before playback can start. There are two ways that the prepared state can be entered:

1. Synchronous way using the `prepare ()` method
2. Asynchronous way using the `prepareAsync ()` method

The difference between those methods is in what thread they are executed. The `prepare ()` method runs in the UI (user interface) thread and thus takes a long time. It will block your UI thread and users might get an ANR (application not responding) message. The `prepareAsync ()` method, on the other hand, runs in a background thread, and thus the UI thread is not blocked; however, the `MediaPlayer` object might not prepare instantly. Therefore, you want to set `onPreparedListener ()` in order to know when the `MediaPlayer` is ready for use. The `prepareAsync ()` method is generally used for playing live data over streams. This is why the current app uses the `prepareAsync ()` method. It allows playing without blocking the main thread.

4.6 Power Manager and Wake Lock

If the phone goes into a low-power state, it will prevent apps from running. To control the power state on device, you need to use power management to [4]:

1. Keep the CPU running
2. Prevent screen dimming or going off
3. Prevent the backlight from turning on

The Android `WifiLock` [8] class allows an application to keep the Wi-Fi component awake. Acquiring a `WifiLock` will keep the connection on until the app releases the lock. In the app, we have decided to keep the radio stations running even when the device screen is off; hence, we acquired the `WifiLock`.

The radio station names, URLs, and icon links for the preset list of radio stations are all saved or referenced at this component. This enables easy extensions and changes. For example, if you change a streaming URL or icon, you only change it here without the need to change any other parts of the code. Similarly, if you want to add a new station to your list, you simply add the new station to the list of existing stations at this component. There is no need to change other parts of the code.

4.7 User Permissions

To run a background service, you need to declare it inside the *manifest* file. Like the main method in a Java class, the manifest file is the entry point of the app. The app activities as well as proper permissions need to be added to the manifest file as well. Part of the manifest file is shown in Table 1 where the app uses the Internet, wake lock, and read and write access to the app directory permissions. Upon running the app, users must grant these permissions for the app to run. The code snippet below shows these steps for the Android app.

Table 1 App's manifest showing the service and permission declaration

```

<?XML VERSION="1.0" ENCODING="UTF-8"?>
<MANIFEST XMLNS:ANDROID="http://schemas.android.com/apk/res/android ..>
<USES-PERMISSION ANDROID:NAME="ANDROID.PERMISSION.INTERNET" />
<USES-PERMISSION ANDROID:NAME="ANDROID.PERMISSION.WAKE_LOCK" />
<USES-PERMISSION ANDROID:NAME="ANDROID.PERMISSION.READ_EXTERNAL_STORAGE" />
<USES-PERMISSION ANDROID:NAME="ANDROID.PERMISSION.WRITE_EXTERNAL_STORAGE" />
<APPLICATION
...
<SERVICE
  ANDROID:NAME=".RadioService"
  ANDROID:ENABLED="TRUE"
  ANDROID:DESCRIPTION="@STRING/RUNNINGRADIO"
  ANDROID:EXPORTED="FALSE">
</SERVICE>
</APPLICATION>
</MANIFEST>

```

```

@Override
public int onStartCommand(Intent intent, int flags, int startId) {
    super.onStartCommand(intent, flags, startId);
    try {
        ...
        player.setOnPreparedListener(this);
        player.setDataSource(RadioMainActivity.url);
        player.setWakeMode(getApplicationContext(),
            PowerManager.PARTIAL_WAKE_LOCK);
        ...
    } catch (IllegalArgumentException e) { ...}
    return START_STICKY;
}

```

5 Apps Class Structure

The class structure of the instance app is presented in Fig. 4. Using a reverse engineering process, the structure is generated from the code using Android Studio and PlantUML plugin [8]. The diagram reveals that the structure, component, and relation between components match the framework architecture but have additional implementation details:

1. The `RadioService` is a service, i.e., a subclass of the Android service, and implements both `onPrepared` and `onError` listeners.
2. The `MainActivity` is the app interface and implements the listener interface to handle user interactions with the app, i.e., the item selected from the list.
3. The power manager, `WifiLock`, and thread components are inner class members of the radio service.
4. The relationship between components is revealed as a package.

You may note that the permissions are not shown in the class structure diagram. This is because they are included in the Android manifest XML file which is not translated to the class when generating class structure from the code.

6 Conclusion and Future Work

We have presented a preliminary framework for developing custom live streaming multimedia apps for mobile devices, i.e., we identified the principal characteristics of mobile live streaming apps and their components, components' interactions, and the design decision needed to create such apps. The framework is an app architecture that can be adapted to create apps that meet specific requirements. To demonstrate how the generic framework can be used, we presented an app for live streaming audio for Android devices using URLs. The app acts as an instance of the generic

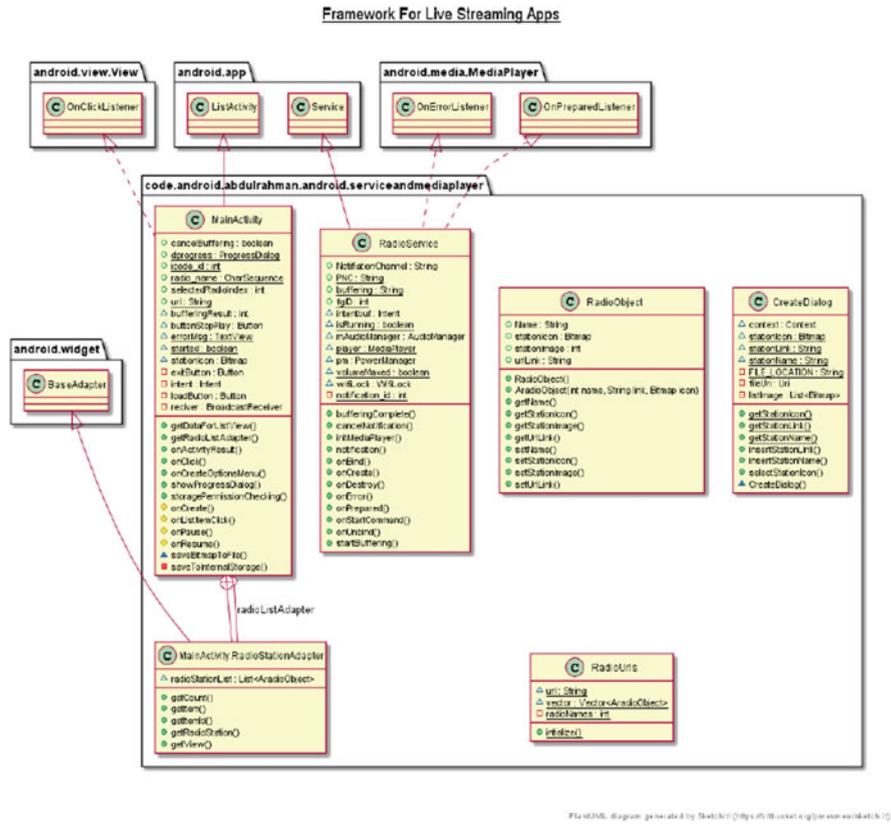


Fig. 4 Class diagram for live streaming radios app

framework and validates it. The app is beneficial to both developers and end users. The paper’s main contributions are (1) the generic framework which can be reused or adapted when developing live streaming multimedia apps or similar ones and (2) using new URLs, customizable live streaming apps can be created. Furthermore, the app’s source code and complete documentation can be used by instructors to teach various Android topics.

The framework is a preliminary one. More work needs to be done to identify and describe properties for individual components and cardinalities between components. A framework-based app can be created for iPhones as well. We will use the knowledge gained from the identified feature works to revise the current preliminary framework and develop a more complete one.

References

1. A. Mawlood-yunis, A live streaming app for android devices, in *2019 International Conference on Computational Science and Computational Intelligence (CSCI), Las Vegas, NV, USA, (2019)*, pp. 1103–1106. <https://doi.org/10.1109/CSCI49370.2019.00209>
2. MediaPlayer Overview, Retrieved May 11, 2020, from <https://developer.android.com/guide/topics/media/mediaplayer>
3. Power management. Retrieved May 11, 2020, from <https://developer.android.com/about/versions/pie/power>
4. WifiManager. Retrieved May 11, 2020, from <https://developer.android.com/reference/kotlin/android/net/wifi/WifiManager>
5. Notification overview. Retrieved May 11, 2020, from <https://developer.android.com/guide/topics/ui/notifiers/notifications>
6. Audio focus. Retrieved May 11, 2020, from <https://developer.android.com/guide/topics/media-apps/audio-focus>
7. <https://www.theguardian.com/media-network/media-network-blog/2013/jan/31/mobile-changing-face-broadcast>
8. <https://www.marketwatch.com/press-release/global-live-streaming-market-research-report-2019-2026-industry-share-and-size-by-value-and-volume-2019-11-28>
9. F. Bentley, D. Lottridge, Understanding mass-market mobile TV behaviors in the streaming era, in *CHI '19: Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, (2019), Paper No.: 261 Pages 1–11, <https://doi.org/10.1145/3290605.3300491>

Change Request Prediction in an Evolving Legacy System: A Comparison



Lamees Alhazzaa and Anneliese Amschler Andrews

1 Introduction

Software systems provide numerous functionalities and innovative features to businesses and organizations. With the growing complexity of software systems, software engineers are pressured to deliver high quality reliable products with predictable costs of maintenance especially in legacy systems. Legacy systems are software systems that are vital to an organization, but due to their age they may have used outdated techniques. Most legacy systems require frequent updates and maintenance requests in order to cope with changes in business [4].

Software evolution refers to the process of repeatedly updating software systems including requirement changes or integration of parts during development. Requirement changes could be an enhancement of features, adaption of systems for changing hardware or software, or performance improvements [15].

Our motive is to be able to use analytical methods to predict Change Requests (CRs) in an evolving aerospace legacy system. Analytical methods such as Software Reliability Growth Models (SRGM) have been used in defect prediction which do not count system enhancement request.

When dealing with evolving systems Musa et al. [15] suggested three main approaches to handle changes in predictions: ignoring change, considering change

L. Alhazzaa

Al-Imam Muhammad Bin Saud Islamic University, Riyadh, Saudi Arabia

e-mail: alhazzaa@imamu.edu.sa

A. A. Andrews (✉)

University of Denver, Denver, CO, USA

e-mail: andrews@cs.du.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_47

671

by re-estimating a new reliability model after each change-point¹ occurrence, and considering changes by applying failure time adjustment after each change-point as if changes were considered since the beginning of a software release (further explanation in Sect. 2).

When applying reliability models to legacy systems, there are issues with the underlying assumptions of SRGM models. Often, these assumptions are violated such as assuming that when defects are fixed no new code was added, or assuming that a given operational profile does not change. This can cause unexpected changes to the defect rate in a system, additionally this can also affect the quality of the defect prediction that could add additional cost for system maintenance and evolution. Stringfellow and Andrews [20] successfully used SRGM models to predict future defects based on data in a defect database (rather than failures as requested by SRGMs). They proposed a selection process to find a candidate model of several models to be used in defect prediction.

This paper contributes in novel ways to use defect prediction methods to help predict Change Requests (CR) in an evolving legacy system. CRs include both corrective and perfective requests with more emphasis on enhancements. Therefore, we address the following research questions:

- RQ1: Can we predict CRs in an evolving legacy system using curve-fitting approaches?
- RQ2: What curve-fitting approaches can we use in CR predictions during evolution and change in legacy systems?
- RQ3: How do these approaches compare?

In this paper we use curve-fitting approaches that are based on SRGM methods to predict future CRs. We use real CR data to compare the prediction accuracy of three different curve-fitting approaches that consider evolution as explained by Musa [15]. We incorporate the idea of Time Transformation (TT) into the curve-fitting approach to demonstrate the use of CR time adjustment when dealing with evolution. Our results show that TT provide more accurate long-term CR predictions than the other approaches.

The remainder of the paper is organized as follows: Sect. 2 provides background on SRGMs in the presence of change-points and available solutions. Section 3 defines our proposed approach, followed by the case study and a discussion of the results in Sect. 4. Conclusions follow in Sect. 5.

Table 1 SRGMs

Model	Equation	Curve shape
Musa or G-O [15]	$\mu(t) = a(1 - e^{(-bt)}) \quad a \geq 0, b > 0$	Concave
Delayed S-shape [22]	$\mu(t) = a(1 - (1 + bt)e^{(-bt)}) \quad a \geq 0, b > 0$	S-shaped
Gompertz [7]	$\mu(t) = a(b^{(c^t)}) \quad a \geq 0, 1 > b > 0, c > 0$	S-shaped
Modified Gompertz [12]	$\mu(t) = d + a(b^{(c^t)}) \quad a \geq 0, 1 > b > 0, c > 0, d > 0$	S-shaped
Yamada exponential [23]	$\mu(t) = a(1 - e^{(-bc^{(1-e^{(-dt)})})}) \quad a \geq 0, 1 > b > 0, c > 0, d > 0$	Concave

2 Background

2.1 Software Reliability Growth Models with Change-Points

SRGM are statistical interpolations of defect detection data by mathematical functions [21]. They are used to predict future defect rates within a software development release. Some of the first software reliability models are the Musa model [15], the delayed S-shaped model [22], the Gompertz model [7], and the Yamada exponential model [23]. In our work we use these four models in addition to a modified version of the Gompertz model called Modified Gompertz [12]. Table 1 contains a summary. Each model gives an equation for $\mu(t)$ which expresses the expected number of failures by time t . The time variable t may be in units of days, weeks, months, etc. The problem with SRGMs is that they do not account for changes in the defect rate. When a change-point occurs, a change in the selected model is required. A change-point is defined as “the point at which the fault detection/introduction rate is changed.” [11]. Changes in a legacy system may occur due to corrective or perfective measures or enhancements. Change-points are estimated using several methods and techniques such as control charts [8, 24], likelihood ratio tests [6, 10, 25], or looking into the number of lines of codes added, deleted, or modified around the time of the change [1].

When dealing with evolving systems, Musa et al. [15] and Lyu [13] highlighted three main approaches to handle system evolution:

¹Change-point is a term used to refer to the change in the defect rate of a software release due to evolution.

1. Ignore change, by performing continual re-estimation of parameters when the current model fails to provide acceptable defect prediction. This approach is used when the total number and volume of changes is small.
2. Apply changes in the model after a change-point by dividing the defect dataset into stages. When a change occurs the dataset after the change is considered a separate dataset where reliability modeling is performed separately. This method considers each stage as an independent component.
3. Apply failure times adjustment. It is most appropriately used when a software is rapidly changing and if it cannot be divided into separate stages.

Each of these approaches has its own pros and cons and they might perform in some systems and databases better than others. These methods are demonstrated and applied later in this paper in Sects. 3 and 4.

2.2 Modeling Approach

We can divide modeling approaches into analytical approaches and curve-fit approaches when using SRGM. Analytical approaches derive a solution analytically by providing assumptions regarding failures, failure repair, and software use and then developing a model based on these assumptions. Curve-fit approaches select a model based on the best curve-fit with few or no assumptions. Curve-fit approaches rely entirely on empirical curve-fitting using one or more types of functions. Both approaches are seen in the literature for software reliability. One of the major contributions for curve-fitting approaches is an empirical study performed by Stringfellow and Andrews [20]. They performed a curve-fitting approach on defect data. This method was not concerned with evolving systems though and no change-points were considered. Chi et al. [5] proposed a multi-stage model that segregates release times based on change-points.

2.3 Defect Prediction vs. Change Request Prediction

On a mapping study of software reliability growth models that consider evolution, by Alhazzaa and Andrews [2], literature shows many studies are concerned with defect prediction and using defect data in evolving software systems. The existing studies do not use databases of Change Requests (CR) which include defects and maintenance requests although many modern software engineering databases contain them.

In addition, many studies focus on finding solutions in terms of Goodness-Of-Fit (GOF). The predictive ability for the proposed solutions is measured for short-term predictions, i.e., one or two time units into the future. Rana et al. [18] and Park et al. [17] highlighted the issue of limited long-term prediction in research. Andrews et

al. [3] used a month-by-month interval to evaluate prediction capabilities for future incident prediction for a help desk rather than CR prediction for software. Since longer-term prediction is a major concern in this work, we will try to adopt this method.

3 Approach

We are looking into three curve-fitting approaches to predict CRs when change-points exist. According to Musa et al. [15] there are three approaches to deal with change, explained in detail in Sect. 2. We demonstrate the first approach, Approach 1, which ignores changes in CR rate Sect. 3.1. Then we show how Approach 2 deals with changes using the multi-stage method in Sect. 3.2. Finally, Approach 3 deals with change by adjusting CR time using Time Transformation (TT) in Sect. 3.3. Our purpose is to find the approach that provides the most accurate predictions with the least amount of under-predicted values. Under-prediction is risky for management, when more CRs occur than they predicted. We describe three curve-fit approaches for SRGM estimation. We will then use these three approaches in our case study for CR prediction and compare their predictive ability.

3.1 Approach 1: Curve-Fitting Approach

This approach uses a cumulative number of CRs over a time period to find a fitted model among several SRGM candidates. When a model is selected it is then used to predict CRs for the remainder of the release. This process was first proposed by Stringfellow and Andrews [20] to predict defects. Based on this approach we use the SRGM in Table 1 to select a model that best fits the cumulative number of CRs. When parameters of a model are estimated, it is evaluated according to the following: The GOF should be of R^2 and should be greater than or equal 0.90. The prediction stability where the estimated value for a week should be within 10% of the estimated value for the previous week. The prediction ability by checking the relative error in $RelativeError = (estimated - actual/actual)$.²

To apply the curve-fitting method to our system we use the process shown in (a) in Fig. 1. After collecting cumulative CRs each week t , the curve-fit program estimates model parameters by attempting to fit the model to the data. If a fit cannot be performed, due to the model not being appropriate for the data or due to insufficient data, then the model is rejected. A sufficient number of data points

²Relative error value is calculated using the absolute value of an error over the actual value. In our case we need to keep track of negative values, therefore we calculated the relative error with the real error value instead.

are determined subjectively. Most curve-fitting tools require at least five data points for the tool to start estimating model parameters and fitting them to the existing data. Some managers might decide that a lower accuracy prediction with a small data set is better than no prediction.

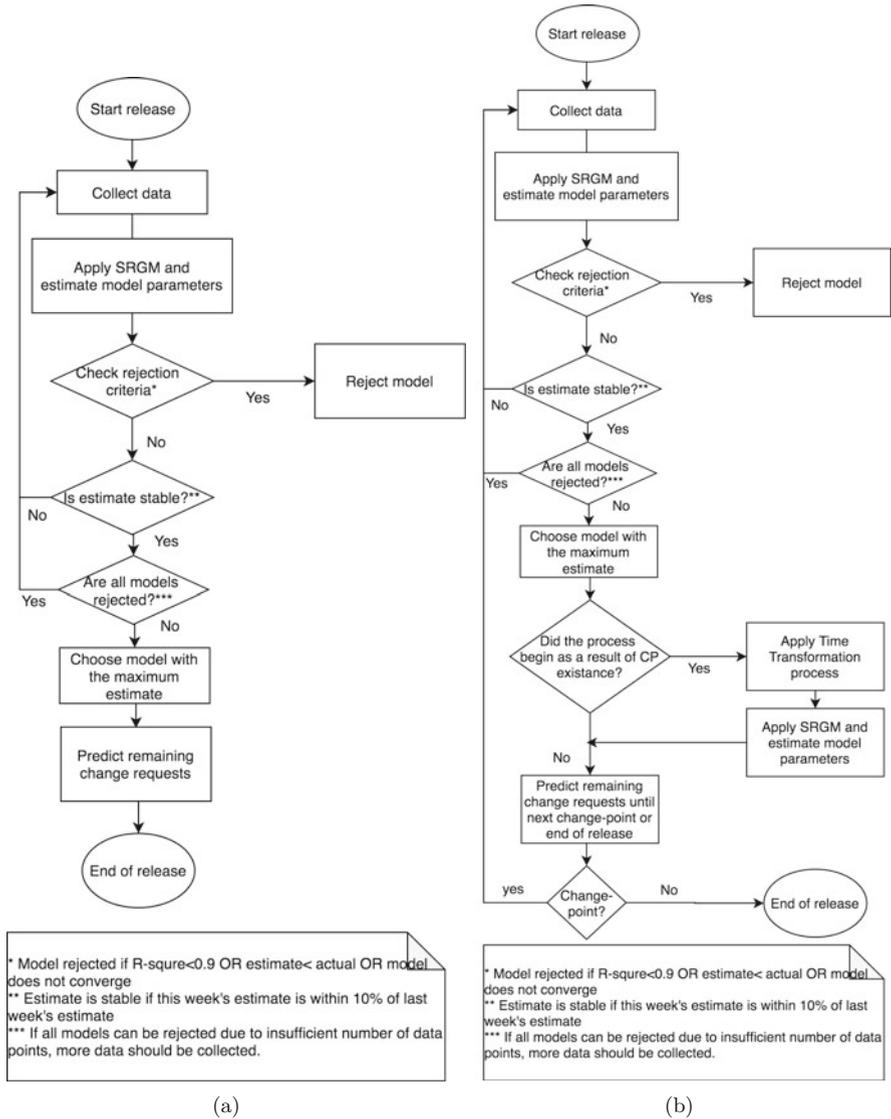


Fig. 1 Model selection and CR estimation using Approach 1 vs. Approach 3. **(a)** Model selection and CR estimation using Approach 1. **(b)** Model selection and CR estimation using Approach 3

If a model's predictions for expected number of total CRs are lower than the actual number of CRs already found and have been consistently so in prior weeks, the model chosen is inappropriate for the data and should not be used in future weeks. If used, it would under-estimate the number of remaining CRs and give a false sense of security. If there is at least one stable model, then the model with the highest R^2 value is chosen for CR prediction.

This approach does not take into consideration the existence of change-points, which can affect the quality of the predictions. Changes in a software system can change the rate of CRs occurring which affects estimation of future CRs.

3.2 Approach 2: Multi-Stage Approach

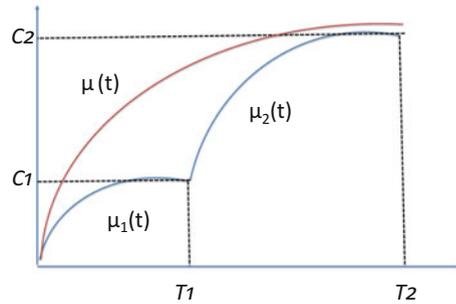
The multi-stage approach was applied by Chi et al. [5] to a defect database. Although the effectiveness of the predictions has not been discussed thoroughly in their work, we find the solution to be interesting to apply to our CR data in order to avoid poor predictions when change-points occur.

For the multi-stage approach we use the same curve-fitting approach in Sect. 3.1 after each change-point, i.e., if a model is selected to perform predictions and a change-point occurs, we are required to fit a new model. After each change-point, we use the curve-fitting approach in Fig. 1a to estimate a new model as if the data after change was in a separate release. This method assumes that a dataset is divided into stages. Each stage has its own fitted model for CR prediction. Change-points were estimated using number of lines of code described in [1].

Let S be a dataset of the cumulative number of CRs in a release over time. This dataset has a number of change-points n . Change-points divide the dataset into $n + 1$ stages, where each stage is referred to as s_i , and $1 \leq i \leq n + 1$. A change-point exists at time T_i , where the total number of CRs for the i^{th} stage is C_i . For each stage s_i , a reliability model is selected $\mu_i(t)$. When a change-point is found at time T_i , a new model is estimated for the next stage. Model $\mu_{i+1}(t)$ will be then used starting at s_{i+1} for CR prediction. The process repeats for each stage until the end of the release.

This method overcomes the issues of selecting a single model in the curve-fitting method in Sect. 3.1 [20]. A disadvantage of this method is that it does not consider each stage as a part of a whole release. This might affect the accuracy of CR predictions. When stages are short, there may not be enough data to select a model and determine parameters according to the selection criteria in Fig. 1a.

Fig. 2 Multi-stage model transformation



3.3 Approach 3: Multi-Stage Approach with Time Transformation

Time Transformation (TT) is a time adjustment method. It overcomes the issue with the multi-stage approach presented in Sect. 3.2 of discarding data from a release prior a change-point. The idea of TT was introduced by Musa et al. [14, 15]. It transforms the defect times after a change to account for code changes as if they existed from the beginning of the release. According to Musa and Iannino “The key principle in developing an adjustment procedure is to modify the failure intervals to the values they would have had for a stable program in its final configuration with complete inspection” [14]. The problem in evolution in cumulative CRs is that when a significant amount of code is changed, the rate of cumulative CR growth changes. In the multi-stage method proposed by Chi et al. [5], we would discard any CR data before the change and we would start all over again after a change-point as if it was a separate release. Approach 3 accounts for the whole release. Before a change-point, the growth rate of cumulative number of CR is different than the growth rate afterwards. TT calculates a model using the new transformed time, which is calculated using the cumulative CR rate using the parameters of the model before the change-point and the parameters of the model after the change. Typically adding a significant amount of code should increase the CR rate.

When the idea of time transformation was proposed by Musa et al. in [14, 15] it was proposed on an analytical model using the same model type before and after change. We plan to build a heterogeneous curve-fit approach that can use a combination of different models to provide the current TT model. In addition, Musa et al. [14, 15] used the model on failure data; we use TT on CR database which is different than failure database.

To explain the process shown in Fig. 1b. The approach starts similar to Approach 1, with the addition of TT after a change-point is detected. When a model is selected after a change-point, TT is performed and new parameters are determined for the new model before using it for CR prediction.

Our goal is to transform $\mu_i(t)$ to $\mu(t)$, where $\mu(t)$ is the curve after TT, see Fig. 2. Let $\mu_i(t)$ be the model selected initially using the curve-fitting approach. At

T_i changes in code are applied and the CR detection rate changes. T_i is the change-point for stage s_i , where i is the number of change-points in the release, $1 \leq i \leq n$, and n is the total number of stages.

- s_1 represents the stage before the first change-point T_1 .
- s_2 represents the stage after the first change-point T_1 and before the second change-point T_2 .
- $s_{(n+1)}$ represents the last stage after the last change-point.

To perform TT on $\mu_i(t)$ to produce the model $\mu(t)$ we calculate transformation time t^* for each time unit j in the timeline, $1 \leq j \leq m$, m being the total number of weeks in the software release. For each stage let C_i represent the total number of cumulative CRs in stage i that occurred at time T_i . Stage 1 has C_1 cumulative CRs which were found by week T_1 , while stage 2 has $C_2 - C_1$ cumulative CRs which were found in weeks $T_1 + 1$ to T_2 .

To perform TT on the data up to T_2 , let $\mu_1(t)$ be the model selected until T_1 and let $\mu_2(t)$ be the model selected after the first change-point according to Approach 2. We need to transform the time according to $\mu_1(t)$ and derive a transformed version of $\mu_2(t)$, to obtain the model $\mu(t)$. Let

$$\mu_1(t) = \lambda(t) \quad (1)$$

$$\mu_2(t) = \alpha(t) \quad (2)$$

We calculate translated time for CRs before the change \hat{t}_j for $\mu_2(t)$. We assign $\mu_2(t)$ to $\mu_1(t)$ to get the value of the translated time

$$\hat{t}_j = \alpha^{-1}(\lambda(t_j)) \quad (3)$$

We then calculate the expected amount of time τ it would have taken to detect C_1 CRs if the new code was part of the original code. By assigning

$$C_1 = \mu_2(\tau) \quad (4)$$

$$C_1 = \alpha(\tau) \quad (5)$$

$$\tau = \alpha^{-1}(C_1) \quad (6)$$

To calculate translated time for CRs observed after the insertion of the new code, we start by asking the question how much time is required for the new model to observe C_1 CRs? Then all CR times between T_1 and T_2 are transformed using the equation below:

$$t^* = \hat{t} - (T_1 - \tau) \quad (7)$$

The value of τ is less than T_1 . For times $t > T_1$, the transformed data consist of the observed CR counts at the translated times. Finally, the new curve $\mu(t)$ is calculated using the new, transformed data.

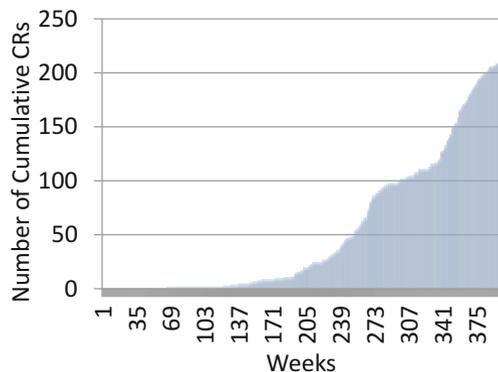
4 Case Study

4.1 Case Study Settings

For the subject system specification, we use data from a release of an aerospace system. It is a legacy system that has been in use for three decades. It has over 1.2 million of lines of code, with most of the code written in C, C++, Java, and scripted code. There are over 850 components in 23 subsystems. The subject system evolved over time. Maintenance included corrective maintenance, adaptations (e.g., to a new hardware), perfective maintenance (e.g., performance improvements), and enhancements with new functionality. Enhancements happen within a single release due to contract obligations. Hence the database accounts for Change Requests (CR). This is very different than a database of failures or defects. In this case we predict a combination of defects and enhancement requests. Each CR has a submission date, the completion date, and the number of hours recorded for CR resolution. LOC added, LOC deleted, LOC modified, and LOC auto-generated for resolving the CRs are recorded as well. In addition, information about the priority of the CR, the functional area where the CR occurs in the system and the type of the request, i.e. if it is a discrepancy or an enhancement.

Data of this study is collected “after the fact,” i.e., it is an ex post facto research. The data were taken from a CR database of an operational release of an aerospace legacy system. Each CR is written to report a problem and is recorded in a CR tracking system (ClearQuest). We grouped data on a weekly basis and then numbered each week. The weeks started from the 34th week of 2008 until the 15th week of 2016. The total number of weeks is 398 weeks, where a total of 211 CRs were found by the end of the release, see Fig. 3. For this release three change-points

Fig. 3 Cumulative number of CRs in the software release



were identified on weeks 247, 265, and 345 according to a previous paper proposed by Alhazzaa and Andrews [1].

IBM SPSS Statistics package [9] is the main software we used to estimate parameters and curve-fit different models. We also used an open source curve-fitting online tool called *MyCurveFit* [16].

4.2 Results

Applying the Curve-Fit Approach We apply curve-fitting according to Approach 1 to the CR database of the case study using the *MyCurveFit* tool. Table 2 shows the weeks where models started fitting data. In week 140, the number of actual CRs is 5. The G-O model estimated only 3 CRs and the R^2 value is only 0.66, which is beneath our threshold, so this model is rejected at this stage. The Delayed S-shaped model estimates only 4 CRs and the R^2 value is only 0.5. The Gompertz model estimates 4 CRs and the R^2 value is 0.81. The Yamada model estimates 3 CRs and the R^2 value is 0.65. And finally the Modified Gompertz estimates 5 CRs which is equal to the actual number of CRs but the R^2 value is only 0.86 which is less than 0.9. The process proceeds to collect data for another week, 141. It rejects all the models as well according to their low R^2 values, which means that more data is collected until week 145. By week 145 the Modified Gompertz model is selected due to its R^2 value meets the minimum threshold requirement 0.9, the number of estimated CRs is equal to the number of actual CRs, and prediction stability is within range since the estimated value for week 145 is within 10% of the value of the previous week. By selecting the Modified Gompertz model we then use it to predict CR in future weeks. Notice that some of the R^2 values gradually change due to gradually adding additional data points. The CR predictions throughout all the stages is shown in Table 6 and it will be further explained in Sect. 4.3.

Applying the Multi-Stage Method Curve-Fit Approach for Change-Points

Using this method, we use the same curve-fitting method we used in Sect. 3.2 to predict CRs for the first stage. We refer to the period before the existence of any change-points as “Stage 1.” When a change-point exists, we start estimating a new curve after the change and the new curve is used then for CR prediction in the future. This method considers the time period after change as “Stage 2.” This applies for multiple change-points, and each time a change-point occurs a new stage is declared. Using the multi-stage method Modified Gompertz is selected for Stage 1 according to Table 2. In week 243, a change in the CR rate occurs. We apply the curve-fitting method for the new stage starting from week 243 we collect data. The minimum number of data points we need to collect to start fitting using our curve-fitting tool is 5 data points. Therefore, we start our first curve-fitting in week 247. In week 247, the Gompertz model has an R^2 value of 0.94 and an estimated value of 48 which matches the actual value, see Table 3. We use this model for CR predictions from this point forward until a change occurs.

Table 2 Estimation using SRGM and the GOF value

Week No.	No. of CRs	G-O		Delayed S-shaped		Gompertz		Yamada		Modified Gompertz	
		Est.	R ²	Est.	R ²	Est.	R ²	Est.	R ²	Est.	R ²
140	5	3	0.66	4	0.5	4	0.81	3	0.65	5	0.86
141	5	3	0.67	4	0.53	4	0.82	3	0.66	5	0.87
142	5	3	0.67	4	0.56	4	0.83	3	0.67	5	0.88
143	5	4	0.68	4	0.59	4	0.83	3	0.67	5	0.89
144	5	4	0.69	5	0.61	4	0.84	4	0.68	5	0.89
145	5	4	0.69	5	0.63	4	0.85	4	0.69	5	0.9

Table 3 Full re-estimation using SRGM and the GOF value for stage 2

Week No.	No. of CRs	G-O		Delayed S-shaped		Gompertz		Yamada		Modified Gompertz	
		Est.	R^2	Est.	R^2	Est.	R^2	Est.	R^2	Est.	R^2
247	48	47	0.44	47	0.74	48	0.94	N/A	N/A	47	0.57

After the second change-point on week 246 a Modified Gompertz model is selected and finally by week 275 using the same approach, see Table 4. After the third change-point in week 357, the R^2 value of both the Gompertz model and Modified Gompertz model is within the acceptable threshold. But these models are rejected due to having the estimated value less than the actual value of cumulative number of CRs. By week 359 all three conditions for selecting a model hold for both Gompertz and modified Gompertz. For this stage the Gompertz model is selected since the d value of the modified Gompertz is equal to zero which makes it a Gompertz model, see Table 5.

Applying the Multi-Stage Method Curve-Fit Approach with Time Transformation for Change-Points This approach starts like the previous curve-fitting approach until a change-point occurs. Then a new curve-fitting is performed to select a new model for the CR data after change. When the new model is selected Time Transformation is performed to adjust the parameters of the final model. After the first change-point, a Gompertz model was selected in a way similar to the multi-stage approach in Sect. 4.2. Time transformation is then applied to the parameters of the Gompertz model to adjust the parameter of the Gompertz model. The new Gompertz model has an R^2 value of 0.94, so it is used to perform predictions of CRs. Likewise after the change-point in week 264, Time transformation is applied to the Modified Gompertz to have GOF of R^2 of 0.97. Finally after the third change-point the Gompertz model used after Time Transformation has an R^2 of 0.93. The resulting model is then used for CR prediction.

4.3 Comparing Predictive Ability

We compare the number of cumulative CRs for every month for a period of 6 months after a model was selected. Approach 1 does not consider change-points [20], Approach 2 starts curve-fitting at each change-point [5], and Approach 3 uses TT at each change-point. In every stage, after model selection, the model is then used for longer-term (6 months) CR prediction. We show the results in Tables 6, 7 and 8. They are structured as follows: The first column of the table shows the last week before prediction. We used week 145 where the model was estimated for the first stage and the weeks after are the weeks where estimation stopped for each of the stages. The second column represents the number of CRs of that specific week. The next column gives the number of predicted CRs after each month, for up to

Table 4 Full re-estimation using SRGM and the GOF value for stage 3

Week No.	No. of CRs	G-O		Delayed S-shaped		Gompertz		Yamada		Modified Gompertz	
		Est.	R ²	Est.	R ²	Est.	R ²	Est.	R ²	Est.	R ²
268	80	72	0.14	73	0.27	74	0.58	72	0.14	76	0.73
269	81	74	0.15	75	0.29	77	0.61	74	0.15	78	0.77
270	83	76	0.16	76	0.31	79	0.67	76	0.16	81	0.81
271	86	77	0.17	78	0.33	81	0.69	77	0.17	83	0.83
272	86	78	0.19	80	0.35	83	0.72	78	0.18	85	0.86
273	87	80	0.2	81	0.38	85	0.76	80	0.2	87	0.88
274	88	81	0.21	82	0.4	86	0.74	81	0.21	89	0.89
275	89	82	0.23	83	0.43	88	0.8	82	0.22	91	0.9

Table 5 Full re-estimation using SRGM and the GOF value for stage 4

Week No.	No. of CRs	G-O		Delayed S-shaped		Gompertz		Yamada		Modified Gompertz	
		Est.	R ²	Est.	R ²	Est.	R ²	Est.	R ²	Est.	R ²
354	154	150	0.34	152	0.6	156	0.88	N/A	N/A	155	0.89
355	154	151	0.37	153	0.65	157	0.8	N/A	N/A	156	0.86
356	159	153	0.37	154	0.65	160	0.87	N/A	N/A	158	0.9
357	165	154	0.33	156	0.59	162	0.9	N/A	N/A	162	0.91
358	166	156	0.33	158	0.58	165	0.93	N/A	N/A	165	0.93
359	167	157	0.33	160	0.59	167	0.94	N/A	N/A	167	0.94

6 months, i.e., (+1 mo.) means predictions after the first month and (+2 mo.) is after 2 months. The last columns record the relative error value. When the relative error equals zero that means that the predicted number of CRs matches the actual number of CRs. When it is negative, it means that the predicted number of CRs is less than the actual number of CRs, which indicates that the model under-predicted the number of CRs and is rejected.

In Table 6 the first row shows week 145, which is the week where the Modified Gompertz model was selected as a model to provide CR predictions. The first 2 months have a relative error of zero, which means predictions are accurate. Afterwards, the relative error ranges from (-0.13) to (0.11) . We then test the predictions of the model after each change-point for 6 months ahead. We find that the range of the relative error varies and can reach a value of 7.64, which is very high compared to the other approaches.

In Table 7, we show the relative error before the first change-point in week 145, which is the same for Approach 1, since no changes in model selection have been made yet. Week 247 is the week where a model was selected for the second stage and prediction started. The model has a relative error value of 0.04. The relative error in the following months ranges from (-0.05) to (-0.26) . As we get further in time, the relative error increases, showing less accurate predictions. After week 275, relative error is 0.8 after 1 month and 0.34 for a 6 month prediction. Finally after the last change-point, the relative error starts with 0.03 after one month to 0.17 after 6 months. In Table 7 we see that the predictions in general have low relative error values compared to Approach 1, especially when performing predictions after change-points. This represents an improvement over Approach 1 results. Table 8 shows the predictions and relative errors after performing TT. We also found that the relative error is relatively low compared to Approach 1 as well. In comparison with Approach 2, TT improves the relative error values. In week 247, the relative error value after 2 months is zero rather than the negative error value if Approach 2 had been applied. We then noticed a decrease of relative error values for every month, which makes this method an improvement in terms of finding better predicted values. Looking into the predictions after week 275 and week 359 all show an improvement of relative error values. We highlighted the relative error values of the monthly prediction in Table 7 in comparison with the relative error in Table 8. The approach that provides higher relative error value among the two approaches, Approach 2 and Approach 3, is highlighted in red and the approach with the lower relative error is highlighted in green. We excluded Approach 1 from this comparison because the relative error values are higher than the other two approaches.

We then revisited our research questions stated in the introduction. RQ1: Can we predict CRs in an evolving legacy system using curve-fitting approaches? From our case study we find that we can use curve-fitting defect prediction approaches in CR prediction. The results are promising in predicting CR similar to predicting defects.

RQ2: What curve-fitting approaches can we use in CR predictions during evolution and change in legacy systems?

Musa et al. [15] provided general guidelines on how to deal with evolution. We considered those guidelines together with adapting them. Stringfellow and Andrews

Table 6 CR predictions and relative errors for 6 months into the future using Approach 1

Week	No. of CRs	No. of Predicted CRs						Relative Error					
		+1 mo.	+2 mo.	+3 mo.	+4 mo.	+5 mo.	+6 mo.	+1 mo.	+2 mo.	+3 mo.	+4 mo.	+5 mo.	+6 mo.
145	5	6	7	7	8	9	10	0.00	0.00	-0.13	-0.11	0.00	0.11
247	48	94	104	115	128	141	155	0.92	0.89	0.92	0.94	0.91	0.80
275	89	188	207	228	250	275	301	1.04	1.16	1.33	1.55	1.81	1.95
359	167	943	1023	1108	1199	1296	1401	5.83	6.06	6.39	6.73	7.13	7.64

Table 7 CR predictions and relative errors for 6 months into the future using Approach 2

Week	No. of CRs	No. of Predicted CRs						Relative Error					
		+1 mo.	+2 mo.	+3 mo.	+4 mo.	+5 mo.	+6 mo.	+1 mo.	+2 mo.	+3 mo.	+4 mo.	+5 mo.	+6 mo.
145	5	6	7	7	8	9	10	0.00	0.00	-0.13	-0.11	0.00	0.11
247	48	51	54	57	61	64	68	0.04	-0.02	-0.05	-0.08	-0.16	-0.26
275	89	100	110	120	131	143	155	0.08	0.13	0.18	0.25	0.31	0.34
359	167	177	187	198	209	221	234	0.03	0.04	0.06	0.09	0.12	0.17

Table 8 CR predictions and relative errors for 6 months into the future using Approach 3

Week	No. of CRs	No. of Predicted CRs						Relative Error					
		+1 mo.	+2 mo.	+3 mo.	+4 mo.	+5 mo.	+6 mo.	+1 mo.	+2 mo.	+3 mo.	+4 mo.	+5 mo.	+6 mo.
145	5	6	7	7	8	9	10	0.00	0.00	-0.13	-0.11	0.00	0.11
247	48	51	55	58	62	66	71	0.04	0.00	-0.03	-0.06	-0.12	-0.21
275	89	99	108	118	129	140	153	0.07	0.11	0.17	0.24	0.30	0.33
359	167	174	183	192	201	211	221	0.01	0.02	0.03	0.04	0.07	0.10

[20] curve-fitting method that was successful in defect prediction by enhancing it to consider change-points. Chi et al. [5] provided a case study that used a multi-staged method that would re-estimate a new model after each change-point. Musa et al. [14, 15] proposed the idea of considering change-points and transforming the model times to consider the whole release. We found that the curve-fitting method provides prediction with low error.

RQ3: How do these approaches compare? When change-points are ignored, CR prediction error increases dramatically as shown in Table 6. Dividing the release into stages and applying the curve-fitting approach provides more accurate results and lower error especially after change-points. The issue with this method is that it discards old data and starts modeling over with new data points. This gives fewer data points to use in curve-fitting, which affects the quality and reliability of the results. Adding a TT step to the existing curve-fitting approach accounts for all the data in the release, and uses curves before and after change to find a third curve that accounts for change as if it had existed from the beginning of the release. By comparing the TT method to the multi-stage curve-fitting method in Tables 7 and 8, we find that the relative error is smaller when TT is applied in all the months except for the third month after week 247 where the multi-stage method provides less relative error. In general we find relative error values are more likely to stabilize or decrease over time when multi-stage or TT approaches are applied compared to the first approach. We also find that the TT approach is superior to the multi-stage approach in providing lower relative error values.

In trying to find what is the best solution, there is no straightforward answer. Each approach is suitable for a specific type of data. If a release has minimal changes that do not affect the CR rate, then Approach 1 would be a suitable approach. When evolution exists the choice is between Approach 2 and Approach 3. Approach 2 provides a simple solution that re-estimates models as required. This is beneficial if at each stage there are enough data points to perform the curve-fitting. It is not recommended to use when change-points are frequent. The problem with this approach is that under-estimation of CRs is likely to occur due to over-fitting. In addition, sometimes a model is selected early in a particular stage based on very few data points. This could lead the curve-fitting tool to settle on a model that poorly predicts future CRs. Approach 3, using TT overcomes the issues in Approach 2. After a change-point, when a model is selected, TT includes data from the beginning of the release to estimate the new model parameters and this overcomes the risk of curve-fitting with too little data. TT also reduces the risk of over-fitting models and causing under-estimation. In CR prediction, a model that frequently under-estimates CRs is not desirable and introduces the risk of management not being prepared for the number of CRs in the future. So we find that TT is a good fit in an evolving release to provide both short-term and longer-term CR prediction. In industry there are many systems that evolve during a release for a variety of reasons, our aerospace system is one of them.

4.4 *Validity Threats*

We follow the guidelines by Runeson et al. [19] in defining our validity threats. An external validity threat is concerned with the generalization of our results. Although we used our method on an evolving system we do not claim that it will produce similar outcomes for other software systems. Our data was collected by a third party which means the researchers have less control over the quality of the data, which is a threat to internal validity. Construct validity refers to the relation between theory and observation. We observed that some models fit data better than others. This may be affected by the number of data points used for model estimation and selection. Although we do not claim that there is a linear relationship between the amount of data and the accuracy of prediction but if the size of data used is too small we may be at risk of selecting a model that does not predict very well.

5 Conclusion and Future Work

In this case study, we investigate the use of three different curve-fitting approaches that have been used in defect prediction for predicting Change Requests (CR) instead. We tested their ability to predict future CRs using a CR database of an evolving aerospace legacy system. We then compared the predictive ability of each of the curve-fitting approaches in an effort to find the approach with the most accurate prediction of CRs. The predictions were performed monthly for up to 6 months after a model was selected. We applied the first approach [20], curve-fitting approach that does not account for change-points. The predictions showed a low relative error at first, but as soon as the release evolved, the predicted number of CRs was much higher than the actual number of CRs. The second approach applied was a multi-stage approach that segments the dataset whenever a change-point is found. The multi-stage model based on the work presented by Chi et al. [5] had proven to give lower relative error in the predicted values but many times the values are underestimated. Under-estimation of the number of CRs puts an organization at risk for not being prepared for the volume of CRs. Finally, the use of Time Transformation (TT) first proposed by Musa et al. [14, 15] along with the curve-fit approach has shown predictions with lower error rate than both of the other approaches and with fewer under-predicted values. The idea of TT has not been widely used in literature. In fact it was demonstrated only with analytical, homogeneous models. The assumptions upon which these models are based on are not met when CRs are considered. For industrial change databases that contain CRs for both defects and enhancements, curve-fit methods are more realistic since they only select an appropriate model according to the given dataset without the assumptions made by analytic models. As future work we plan to extend our case study and compare results among multiple releases. This would show the generalizability of our results.

References

1. L. Alhazzaa, A. Amschler Andrews, Estimating change-points based on defect data, in *The 2018 International Conference on Computational Science and Computational Intelligence (CSCI)* (IEEE, New York, 2018), pp. 878–883
2. L. Alhazzaa, A. Amschler Andrews, A systematic mapping study on software reliability growth models that consider evolution, in *Proceedings of the International Conference on Software Engineering Research and Practice (SERP)*. The Steering Committee of the World Congress in Computer Science, Computer ... (2019), pp. 83–90
3. A.A. Andrews, P. Beaver, J. Lucente, Towards better help desk planning: predicting incidents and required effort. *J. Syst. Softw.* **117**, 426–449 (2016)
4. K. Bennett, Legacy systems: coping with success. *IEEE softw.* **12**(1), 19–23 (1995)
5. J. Chi, K. Honda, H. Washizaki, Y. Fukazawa, K. Munakata, S. Morita, T. Uehara, R. Yamamoto, Defect analysis and prediction by applying the multistage software reliability growth model, in *2017 8th International Workshop on Empirical Software Engineering in Practice (IWESEP)* (IEEE, New York, 2017), pp. 7–11
6. P. Galeano, The use of cumulative sums for detection of change-points in the rate parameter of a Poisson process. *Comput. Stat. Data Anal.* **51**(12), 6151–6165 (2007)
7. B. Gompertz, On the nature of the function expressive of the law of human mortality, and on a new mode of determining the value of life contingencies. *Philos. Trans. R. Soc. Lond. Ser. A* **115**(513), 252–253 (1825)
8. C.Y. Huang, T.Y. Hung, Software reliability analysis and assessment using queueing models with multiple change-points. *Comput. Math. Appl.* **60**(7), 2015–2030 (2010)
9. IBM Corp.: IBM SPSS statistics for windows (2017). <https://www.ibm.com/products/spss-statistics>
10. S. Inoue, S. Hayashida, S. Yamada, Toward practical software reliability assessment with change-point based on hazard rate models, in *2013 IEEE 37th Annual Computer Software and Applications Conference (COMPSAC)* (IEEE, New York, 2013), pp. 268–273
11. M. Jain, T. Manjula, T. Gulati, Software reliability growth model (SRGM) with imperfect debugging, fault reduction factor and multiple change-point, in *Proceedings of the International Conference on Soft Computing for Problem Solving (SocProS 2011)*, December 20–22, 2011 (Springer, New Delhi, 2012), pp. 1027–1037
12. S. Jiang, D. Kececioglu, P. Vassiliou, Modified Gompertz, in *Proceedings of the Annual Reliability and Maintainability Symposium* (1994)
13. M.R. Lyu et al., *Handbook of Software Reliability Engineering* (1996)
14. J.D. Musa, A. Iannino, Software reliability modeling: accounting for program size variation due to integration or design changes. *ACM SIGMETRICS Perform. Eval. Rev.* **10**(2), 16–25 (1981)
15. J.D. Musa, A. Iannino, K. Okumoto, *Software Reliability: Measurement, Prediction, Application* (McGraw-Hill, New York, 1987)
16. MyAssays Ltd.: MyCurveFit online curve fitting. <https://mycurvefit.com/>
17. J. Park, N. Lee, J. Baik, On the long-term predictive capability of data-driven software reliability model: an empirical evaluation, in *2014 IEEE 25th International Symposium on Software Reliability Engineering* (IEEE, New York, 2014), pp. 45–54
18. R. Rana, M. Staron, C. Berger, J. Hansson, M. Nilsson, F. Törner, Evaluating long-term predictive power of standard reliability growth models on automotive systems, in *2013 IEEE 24th International Symposium on Software Reliability Engineering (ISSRE)* (IEEE, New York, 2013), pp. 228–237
19. P. Runeson, M. Höst, Guidelines for conducting and reporting case study research in software engineering. *Empir. Softw. Eng.* **14**(2), 131 (2009)
20. C. Stringfellow, A.A. Andrews, An empirical method for selecting software reliability growth models. *Empir. Softw. Eng.* **7**(4), 319–343 (2002)
21. A. Wood, Software reliability growth models. Tandem technical report 96(130056) (1996)

22. S. Yamada, M. Ohba, S. Osaki, S-shaped reliability growth modeling for software error detection. *IEEE Trans. Reliabil.* **32**(5), 475–484 (1983)
23. S. Yamada, H. Ohtera, H. Narihisa, Software reliability growth models with testing-effort. *IEEE Trans. Reliabil.* **35**(1), 19–23 (1986)
24. Y. Zhao, C. Wan, F. Gao, S. Chang, Change points estimation and simulation for software reliability model, in *2013 International Conference on Measurement, Information and Control (ICMIC)*, vol. 1 (IEEE, 2013), pp. 434–438
25. F.Z. Zou, A change-point perspective on the software failure process. *Softw. Test. Verif. Reliabil.* **13**(2), 85–93 (2003)

Using Clients to Support Extract Class Refactoring



Musaad Alzahrani

1 Introduction

Software systems that have a long lifetime (e.g., enterprise systems) usually undergo evolutionary changes in order to remain useful due to various reasons including changes to business rules, user requirements, hardware, and platforms. Unfortunately, most of the programmers who are responsible for making these changes in a system modify the source code of that system rapidly without considering the resultant effects on the design of the system. As a result, the design quality of the system deteriorates; and the system becomes very difficult to understand and change.

One of the most common design issues in object-oriented systems is having a class with many responsibilities. During the maintenance and evolution activities of a system, new responsibilities may need to be added to the system. Due to time limits, software developers usually feel that it is not necessary to create a separate class for a new responsibility and that responsibility can be added to one of the existing classes [1]. After several cycles of maintenance and evolution, some classes in the system will end up having many responsibilities which will increase the maintenance cost of the system because a class with many responsibilities requires more effort and time to understand and maintain. In addition, a class with many responsibilities has many reasons to change because each responsibility the class has is an axis of change [2]. Such a class needs to be refactored in order to improve its understandability and maintainability. The refactoring technique that is usually applied to overcome this issue is called Extract Class refactoring. It refers to the

M. Alzahrani (✉)
Albaha University, Albaha, Saudi Arabia
e-mail: malzahr@bu.edu.sa

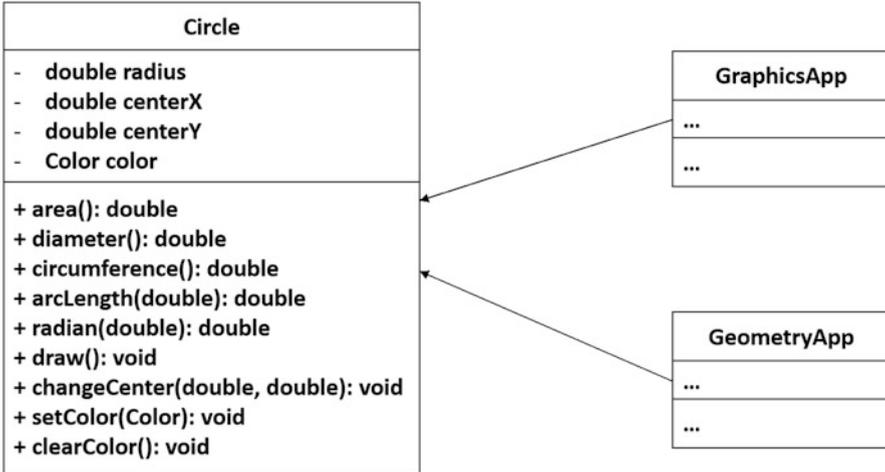


Fig. 1 The class Circle and its clients

process of splitting a class that has many responsibilities into a set of smaller classes, each of which has one responsibility and one reason to change [1].

Manually performing the process of the Extract Class refactoring costs much time and effort. Thus, several approaches (e.g., [3–5]) have been introduced in the literature to automate and support the Extract Class refactoring. These approaches conduct the Extract Class refactoring based on factors internal to the class to be refactored such as structural and semantic dependency between the methods of the class. However, this internal view of the class is inadequate in many cases to automatically determine the responsibilities of the class and to determine the potential reasons that can cause changes to the class. For instance, consider the class *Circle* shown in Fig. 1. The class has four attributes and nine methods. In addition, the class has two client classes: *GeometryApp* and *GraphicsApp*. We refer to these two classes as clients of the class *Circle* because they use methods in the class (see Sect. 3). The class *GeometryApp* performs some computational geometry. Thus, it uses the methods that provide geometric computations (e.g., *area()*) in the class *Circle* but never uses the methods that provide drawing services (e.g., *draw()*). The other client class (i.e., *Graphics*) uses the methods that provide drawing features in the class *Circle* because it draws shapes on screen including circles.

It is obvious that the class *Circle* has two responsibilities: one responsibility is to perform computational geometry and the other responsibility is to perform drawings on the screen. These two responsibilities may increase the cost and effort of future changes of the class *Circle*. Assume that the client class *Graphics* exerts a change on the method *setColor(Color)* in the class *Circle*. This change may affect the client class *Geometry* because it depends on the class *Circle*. Therefore, the class *Geometry* may need to be recompiled and retested even though it never uses the

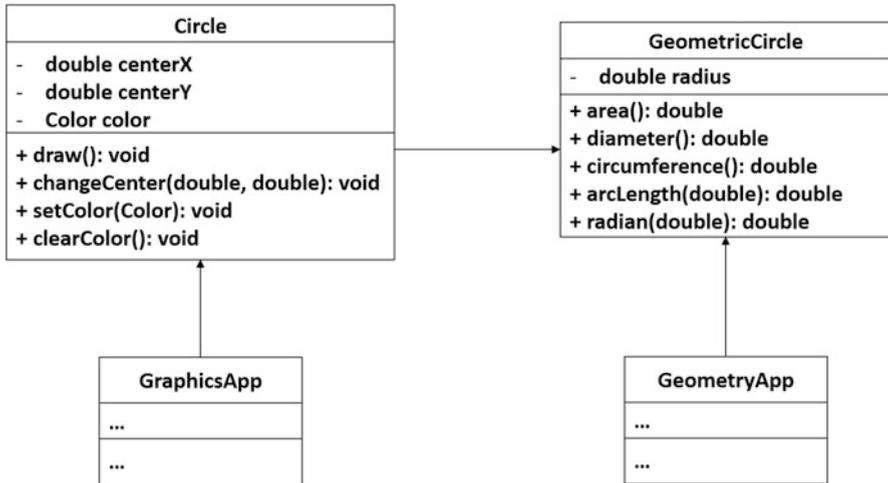


Fig. 2 The class `Circle` in Fig. 1 is split into two classes: `Circle` and `GeometricCircle`

method `setColor(Color)`. In order to address this issue, we need to perform the Extract Class refactoring by splitting these two responsibilities into two separate classes as shown in Fig. 2. Existing approaches of the Extract Class refactoring may fail to suggest the refactoring solution shown in Fig. 2 because they perform the Extract Class refactoring based on structural and semantic relationships between the methods of the class. The structural relationships between methods in these approaches are measured based on the shared attributes between the methods, and the semantic relationships are measured based on the common vocabularies in the documents (i.e., identifiers and comments) of the methods. Thus, most of the methods of the class `Circle` are (to some degree) structurally and semantically related to one another because they likely share attributes and vocabularies (e.g., `radius`).

To overcome the above issue, this study proposes a novel approach that performs the Extract Class refactoring based on the similarities in terms of clients between the methods of the class in question. The proposed approach identifies the different responsibilities of a class based on the usage of its methods by its clients. The intuition behind this is that if there are different sets of methods in the class that are used by different sets of clients, then the class has different responsibilities from the perspective of its clients [6–8].

The proposed approach could be potentially more beneficial than the traditional refactoring techniques that consider only the internal view of the class when performing the extract class refactoring because it also supports the Interface Segregation Principle (ISP). ISP is an important object-oriented design principle that states “no client should be forced to depend on methods it does not use” [2]. Our proposed approach is not meant to replace the existing approaches of the Extract Class refactoring but to complement them because considering the structural and

semantic relationships between the methods of class can be useful in many cases of the Extract Class refactoring.

The rest of the chapter is organized as follows. In Sect. 2, we present and discuss the related work. Section 3 presents the proposed approach. Section 4 gives the conclusion and future work.

2 Related Work

Several approaches have been introduced in the literature to try to support and automate the Extract Class refactoring. In the following, we discuss and summarize the approaches that are mostly relevant to our approach.

Fokaefs et al. [3] proposed an approach for the Extract Class refactoring which employs an agglomerative clustering algorithm based on the Jaccard distance between the methods of the class to be refactored. Structural similarity between two methods of the class is used to calculate the Jaccard distance between the two methods. The higher the Jaccard distance between the two methods is, the more probability the two methods will be in the same cluster. The resulting clusters represent the Extract Class opportunities that can be identified in the class to be refactored.

Bavota et al. [4] introduced an approach that splits the class to be refactored into two classes with higher cohesion than the original class-based structural and semantic similarities between the methods of the class. The class is represented as a weighted graph where the nodes of the graph represent the methods of the class and the weighted edges of the graph represent the degree of structural and semantic similarities between the methods of the class. The approach uses the Max-Flow Min-Cut algorithm [9] to split the weighed graph representing the original class into two weighted subgraphs representing the two classes that can be extracted from the original class.

In [5], Bavota et al. introduced another approach that can automatically decompose the class to be refactored into two classes or more. The class is represented as a weighted graph in a similar manner of their previous approach. Instead of using the Max-Flow Min-Cut algorithm, they used a two-step clustering algorithm that removes the edges of the graph that have light weights to split the graph into a set of subgraphs representing the classes that can be extracted from the original class.

Previous approaches of the Extract Class refactoring consider only factors internal to the class to be refactored. These approaches may potentially fail in many cases to identify the ideal Extract Class refactoring opportunities in the class. In this chapter, we introduce a new approach that automatically performs the Extract Class refactoring considering factors external to the class.

3 Extract Class Refactoring Based on Clients

In this section, we first present a set of basic definitions. In addition, we present the proposed approach for the Extract Class refactoring. Furthermore, we present an example to show how our approach can be applied.

3.1 Definitions

Definition 1 (Classes) $S = \{c_1, c_2, \dots, c_n\}$ is the set of classes that composes an object-oriented system.

Definition 2 (Methods of a Class) Let $c \in S$. Then, $M(c) = \{m_1, m_2, \dots, m_k\}$ is the set of methods implemented in the class c .

Definition 3 (Clients of a Method) Let $c \in S$, and $m \in M(c)$. Then, $Client(m) = \{c' \in S - \{c\} \mid \exists m' \in M(c') \text{ and } m' \text{ invokes or may, because of polymorphism, invoke } m\}$ is the set of clients of m .

Definition 4 (Degree of Client-Based Similarity Between Two Methods) Let $c \in S$ and $m_i, m_j \in M(c)$. Then, the degree of client-based similarity between the method m_i and the method m_j is defined by

$$Sim_{clients}(m_i, m_j) = \begin{cases} \frac{|Client(m_i) \cap Client(m_j)|}{|Client(m_i) \cup Client(m_j)|} & \text{if } |Client(m_i) \cup Client(m_j)| > 0, \\ 0 & \text{otherwise} \end{cases}$$

Definition 5 (Degree of Client-Based Similarity Between a Method and a Set of Methods) Let $c \in S$, $m \in M(c)$, $SM \subset M(c)$ and $m \notin SM$. Then, the degree of client-based similarity between the method m and the set of methods SM is defined by

$$Sim_{MethodWithSetclients}(m, SM) = \begin{cases} \frac{\sum_{x \in SM} Sim_{clients}(m, x)}{|SM|} & \text{if } |SM| > 0, \\ 0 & \text{otherwise} \end{cases}$$

3.2 The Proposed Approach for Extracting Classes

The main steps of our approach are shown in Fig. 3. Given a class to be refactored, we first extract the methods of the class. Then, we determine the clients of each method using Definition 3. Once we have the clients of each method in the class, we compute the degree of client-based similarity between each pair methods in the

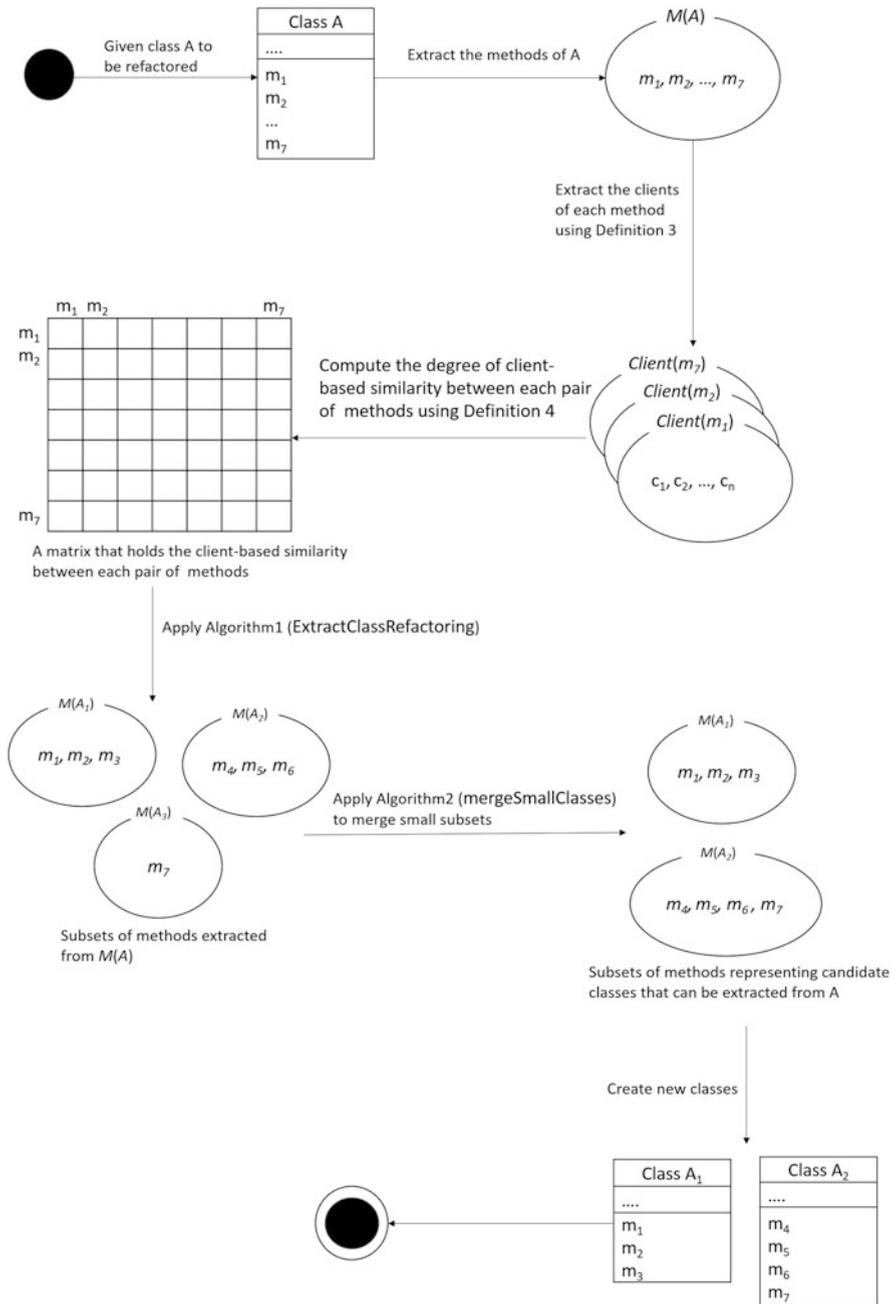


Fig. 3 The main steps of our approach

Algorithm 1: ExtractClassRefactoring($M(c)$, $Threshold$, $k \times k$ matrix)

Input: 1) $M(c)$: the set of methods of the class c to be refactored. 2) $Threshold$: a threshold value. 3) $k \times k$ matrix: holds $Sim_{clients}$ between each pair of methods in the class where k is the number of methods in the class

Output: a clutter F of subsets of $M(c)$ such that each subset represents a class can be extracted from c

```

begin
   $F = \{ \}$ ;
  while  $|M(c)| > 1$  do
    find the pair of methods  $m_i, m_j \in M(c)$  with highest  $Sim_{clients}(m_i, m_j)$  such that
     $i \neq j$ ;
    if  $Sim_{clients}(m_i, m_j) < Threshold$  then
      break;
    else
      add  $m_i, m_j$  to a new  $Subset$ ;
      remove  $m_i, m_j$  from  $M(c)$ ;
      while  $|M(c)| > 0$  do
        find the method  $m_k \in M(c)$  with highest
         $SimMethodWithSetclients(m_k, Subset)$ 
        if  $SimMethodWithSetclients(m_k, Subset) > Threshold$  then
          add  $m_k$  to  $Subset$ ;
          remove  $m_k$  from  $M(c)$ ;
        else
          break;
        end
      end
      add  $Subset$  to  $F$ ;
    end
    while  $|M(c)| > 0$  do
      add each remaining method  $m_r$  in  $M(c)$  to a new  $Subset$ ;
      add  $Subset$  to  $F$ ;
      remove  $m_r$  from  $M(c)$ ;
    end
  end
end
return  $F$ 
end

```

class using Definition 4 and store the results in $k \times k$ matrix, where k is the number of methods in the class. The entry $[i][j]$ of the $k \times k$ matrix holds the degree of client-based similarity between the method m_i and the method m_j .

Algorithm 1 is next applied. The algorithm takes as an input the set of methods of the class to be refactored, the $k \times k$ matrix that holds the degree of client-based similarity between each pair of methods, and a threshold value. The algorithm returns as an output a clutter (i.e., family) of subsets of the input set of methods. The algorithm classifies the input set of methods into different subsets of methods based on the client-based similarities between the methods. The input threshold value is used to determine if a method to be classified is added to an existing subset of methods or added to a new subset. Therefore, if the threshold value is high and

Algorithm 2: mergeSmallClasses(F , $minNumMethods$, $k \times k$ matrix)

Input: 1) F : the output of Algorithm 1 which may include small subsets of methods. 2) $minNumMethods$: the minimum number of methods that each extracted class can have 3) $k \times k$ matrix: holds $Sim_{clients}$ between each pair of methods in the original class where k is the number of methods in the class

Output: F after merging small subsets

```

begin
  for  $X \in F$  do
    if  $|x| < minNumMethods$  then
      find  $Y \in F - X$  that has the highest average of client-based similarities
      between its pairs of methods;
      add the elements of  $X$  to  $Y$ ;
      remove  $X$  from  $F$ ;
    end
  end
  return  $F$ 
end

```

the client-based similarities between the methods of the class to be refactored are generally low, each method will probably be added to a different subset, which means that each extracted class will have only one method. To overcome this issue, the median of the non-zero client-based similarities of all pairs of methods in the class can be chosen as a threshold value.

Finally, Algorithm 2 is applied to avoid the extraction of classes that have a small number of methods (e.g., one method). The algorithm takes as an input the clutter F resulting from Algorithm 1, the minimum number of methods that each extracted class can have, and the $k \times k$ matrix that holds the degree of client-based similarity between each pair of methods. The algorithm returns as an output a clutter of subsets such that each subset has a number of methods that is equal to or more than the input minimum number of methods. Algorithm 2 merges any subset $X \in F$ that has less number of methods than the input minimum number of methods with the subset $Y \in F - X$ that has the highest average of the client-based similarities between pairs of methods in the subset. Each subset in the output of Algorithm 2 represents a candidate class that can be extracted from the class to be refactored.

3.3 Example of Application

We present here an example to show how the proposed approach can be applied. Let class c_1 be the class to be refactored. Let $M(c_1) = \{m_1, m_2, m_3, m_4, m_5, m_6, m_7\}$. Let the clients of each method in c_1 be the following:

- $Client(m_1) = \{c_2, c_3\}$,
- $Client(m_2) = \{c_2, c_3, c_4\}$,
- $Client(m_3) = \{c_3, c_4\}$,

Fig. 4 The client-based similarity between each pair of methods in the class c_1

	m_1	m_2	m_3	m_4	m_5	m_6	m_7
m_1	1	0.67	0.33	0	0	0	0
m_2	0.67	1	0.67	0	0	0	0
m_3	0.33	0.67	11	0	0	0	0
m_4	0	0	0	1	0.25	0.2	0
m_5	0	0	0	0.25	1	0.67	0.33
m_6	0	0	0	0.2	0.67	1	0.67
m_7	0	0	0	0	0.33	0.67	1

- $Client(m_4) = \{c_5, c_6, c_7\}$,
- $Client(m_5) = \{c_7, c_8\}$,
- $Client(m_6) = \{c_7, c_8, c_9\}$,
- $Client(m_7) = \{c_8, c_9\}$.

Given the above sets, we can compute the client-based similarity between each pair of methods of class c_1 using Definition 4. The matrix shown in Fig. 4 holds the client-based similarity between each pair of methods of class c_1 . The entry $[i][j]$ of the matrix holds the value of $Sim_{clients}(m_i, m_j)$.

Algorithm 1 is applied next to perform the Extract Class refactoring on the class c_1 . The algorithm takes three inputs: $M(c_1)$, a *Threshold* value, and the matrix shown in Fig. 4. We set the value of *Threshold* to 0.5, which is the median of the values in the matrix in Fig. 4 excluding the zero values and the main diagonal of the matrix. The output of Algorithm 1 is the following clutter of subsets:

$$F = \{\{m_1, m_2, m_3\}, \{m_4\}, \{m_5, m_6, m_7\}\}.$$

Each subset represents a class that can be extracted from c_1 .

Algorithm 2 can be applied next to avoid extracting classes with a small number of methods. The algorithm takes three inputs: the clutter F (i.e., the output of Algorithm 1), *minNumMethods*, which is a chosen value for the minimum number of methods that each extracted class can have, and the matrix shown in Fig. 4. In this example, we set *minNumMethods* = 2. The following is the output of Algorithm 2:

$$F = \{\{m_1, m_2, m_3\}, \{m_4, m_5, m_6, m_7\}\}.$$

The output of Algorithm 2 shows that the subset $\{m_4\}$ is merged with the subset $\{m_5, m_6, m_7\}$. Thus, two classes are suggested to be extracted from the original class c_1 in our example. The first extracted class has the following set of methods: $\{m_1, m_2, m_3\}$, and the second extracted class has the following set of methods: $\{m_4, m_5, m_6, m_7\}$.

4 Conclusion and Future Work

This chapter introduced a novel approach that performs the Extract Class refactoring. The proposed approach uses the clients of the class to identify classes that can be extracted from the class. The application of the proposed approach was illustrated in the chapter.

In the future work, we plan to conduct a large empirical study that quantitatively analyzes the relationships between a number of approaches for the Extract Class refactoring (including the approach proposed in this chapter) and real cases of Extract Class refactoring for the purpose of identifying the factors that can be used to better identify and separate the different responsibilities of a class.

References

1. M. Fowler, *Refactoring: Improving the Design of Existing Code* (Addison-Wesley Professional, Boston, 1999)
2. R.C. Martin, M. Martin, *Agile Principles, Patterns, and Practices in C# (Robert C. Martin)* (Prentice Hall PTR, Upper Saddle River, 2006)
3. M. Fokaefs, N. Tsantalis, A. Chatzigeorgiou, J. Sander, Decomposing object-oriented class modules using an agglomerative clustering technique, in *2009 IEEE International Conference on Software Maintenance* (IEEE, New York, 2009), pp. 93–101
4. G. Bavota, A. De Lucia, R. Oliveto, Identifying extract class refactoring opportunities using structural and semantic cohesion measures. *J. Syst. Softw.* **84**(3), 397–414 (2011)
5. G. Bavota, A. De Lucia, A. Marcus, R. Oliveto, Automating extract class refactoring: an improved method and its evaluation. *Empir. Softw. Eng.* **19**(6), 1617–1664 (2014)
6. S. Mäkelä, V. Leppänen, Client-based cohesion metrics for java programs. *Sci. Comput. Programm.* **74**(5–6), 355–378 (2009)
7. M. Alzahrani, A. Melton, Defining and validating a client-based cohesion metric for object-oriented classes, in *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1 (IEEE, New York, 2017), pp. 91–96
8. M. Alzahrani, S. Alqithami, A. Melton, Using client-based class cohesion metrics to predict class maintainability, in *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1 (IEEE, New York, 2019), pp. 72–80
9. T.H. Cormen, C.E. Leiserson, R.L. Rivest, C. Stein, *Introduction to Algorithms* (MIT Press, Cambridge, 2009)

Analyzing Technical Debt of a CRM Application by Categorizing Ambiguous Issue Statements



Yasemin Doğancı, Özden Özcan-Top, and Altan Koçyiğit

1 Introduction

Today, most of the vendors delivering software solutions are in a constant competition. Accordingly, the development and implementation efforts are tried to be kept quick, and solutions are delivered rapidly. Technical debt is an outcome of making poor decisions or choosing easier or quicker paths in software development for fast code delivery. Although it may be possible to save time and effort in the short term by this kind of rapidness, introducing technical debt in the product adversely affects the delivered software's quality [1]. Accordingly, in the long term, such technical debt requires more time and effort in terms of maintenance and refactoring. To quantify the degradation in the overall software quality in time, measuring the technical debt using different metrics is crucial.

In the literature, there are studies which compare different technical debt identification methods [2–5], case studies that compare technical debt management in different companies [6–9], research that summarizes the evolution of technical debt [1], and suggestions for ontology of technical debt terms [10].

Although the technical debt concept is widely applicable to any software development project, there is a limited understanding of the causes and effects of it on development efforts in enterprise-level software. Enterprise software definition covers customizable platforms such as enterprise resource planning (ERP), human resources management (HRM), and customer relationship management (CRM). It

Regular Research Paper

Y. Doğancı (✉) · Ö. Özcan-Top · A. Koçyiğit
Graduate School of Informatics, Middle East Technical University, Ankara, Turkey
e-mail: yasemin.doganci@metu.edu.tr; ozdenoz@metu.edu.tr; kocyiigit@metu.edu.tr

is crucial to investigate and identify technical debt in these systems as the success in business highly depends on effective usage of these systems in a well-organized way. Accordingly, the quality of such systems contributes to the overall quality of business processes. Hence, in this study, we specifically focused on technical debt in a CRM platform (i.e., Salesforce¹).

CRM platforms provide a streamlined way of managing businesses by improving collaborative work, presenting easy-to-use features such as reporting tools, and therefore elevating success in businesses [11]. These improvements – either on products or services – are often delivered by software companies that are specialized in platform-based solutions or in-house teams who are responsible for understanding the requirements of business users and building solutions.

According to Suryanarayana et al., “Awareness is the first step toward managing technical debt” [12]. If awareness is high within a project team, then managing technical debt would be easier, by identifying the causes or impacts of the debt.

In this study, we performed a research on identification of the technical debt on the Salesforce platform by using 300 *anonymous issue definitions*. For technical debt categorization, we used three different categorization methods [10, 13, 14] that provide increasing levels of detail. We think that this categorization would help increasing “awareness” among software development teams in CRM platforms.

The issue definitions used in this study were retrieved from an independent software vendor (ISV) (i.e., OrgConfessions²) which provides unbiased confessions of consultants, administrators, and developers from different Salesforce organizations anonymously.

The main research questions of this study are formulated as follows:

1. To what extent we can identify the causes of technical debt based on ambiguous issue statements?
2. How do different technical debt categorization or identification methods help in analyzing the causes of technical debt in a customer relationship management (CRM) platform?

In Sect. 2, we provide a brief background information regarding the technical debt concept and introduce three different technical debt categorization approaches used in this study. We also pointed out related work in Sect. 2. The research methodology comprising data collection and the analysis are presented in Sect. 3. Section 4 presents the validation of the categorization process. Section 5 discusses the technical debt categorization results and Sect. 6 concludes the study.

¹[Salesforce.com](https://www.salesforce.com)

²[OrgConfessions](https://www.orgconfessions.com)

2 Background

2.1 Technical Debt

The term “technical debt” was created as a metaphor by Ward Cunningham [15], to describe all the code defects, and design architectural mistakes made by developers, and to summarize them to “nontechnical” people in software projects. Technical debt occurs when an instantaneous action adds value to the software but leads to undesirable consequences. In other words, taking shortcuts during analysis, design, implementation, testing, or even documentation phases of a project might end up in more effort and time spent on the tasks in order to resolve a defect or to enhance the quality of the end product.

Several technical debt identification methods are suggested by researchers in the literature, and comparisons of these different methods are made since Cunningham’s introduction [2–5, 10, 13, 14]. Among these, we selected (i) Steve McConnell’s [13], (ii) Martin Fowler’s [14], and (iii) Alves et al.’s [10] approach for technical debt categorization. We ordered these three approaches as follows according to the increasing level of granularity of their descriptions and applied them to our context in this order.

Level 1 The first level of technical debt identification can be performed on the “intention” level as suggested by Steve McConnell [13]. In particular, each technical debt can be categorized as “intentional” and “unintentional.” In software projects, technical debts are usually *unintentional*, when imperfect solutions are preferred unconsciously. We discovered that most of the architectural or structural debts fall into this category, since the consequences of the architectural/design decisions could usually be observed at later stages of a software development life cycle.

On the other hand, almost all suboptimal solutions preferred to address the needs of customers or stakeholders *intentionally*, leading to low product quality alongside. Such solutions may be developed to solve design or code defects which blocks the software. Consequences of these actions are mostly known by development teams at the time of the actions and mostly marked as “to be refactored” in the following development iterations. Therefore, such cases are considered as *intentional* technical debt.

Level 2 The second level of technical debt categorization we used is the “technical debt quadrant” developed by Martin Fowler [14]. This quadrant classifies the technical debt based on the intention of the person who creates the debt but in a more detailed way than McConnell’s classification. The classification quadrant shown in Table 1 includes both the classification types and the examples for each classification given by Fowler.

Table 1 The technical debt quadrant of Fowler [14]

	Reckless	Prudent
Deliberate	We don't have time for design	We must ship now and deal with consequences
Inadvertent	What's layering?	Now we know how we should have done it

Below, we provide the explanations of the examples given above:

1. *Reckless – Inadvertent*: “*What's layering?*”. This example refers to the lack of knowledge on good design practices and capability of practicing them as a team of developers. This kind of technical debt is the least desirable one and usually not recognized.
2. *Reckless – Deliberate*: “*We don't have time for design.*” This example may refer to project planning issues and not meeting deadlines, the state of not affording the time required to come up with clean solutions. Quick solutions without proper design, causing long-term defects, are considered mainly in this category.
3. *Prudent – Deliberate*: “*We must ship now and deal with consequences.*” This example refers to meeting certain deadlines with quick and low-quality solutions, but accepting the debt, where the cost of paying it is recognized.
4. *Prudent – Inadvertent*: “*Now we know how we should have done it.*” This example refers to a state where the code or design had been clean but realizing that it could have been designed better to meet the requirements. The debts in this category can be seen as learning opportunities to provide higher quality on upcoming development cycles or efforts.

Level 3 Aside from the *intention* and *carefulness* aspects of technical debt – which were covered by McConnell's and Fowler's approaches, respectively – a more comprehensive taxonomy was formed by Alves et al. [10]. They defined an ontology for the “nature” of the debt, by considering the activities of the development process where the debt occurs. They identified 13 different technical debt types which were correlated with the activity of the development process execution:

- Architecture debt
- Build debt
- Code debt
- Defect debt
- Design debt
- Documentation debt
- Infrastructure debt
- People debt
- Process debt
- Requirement debt
- Service debt
- Test automation debt
- Test debt

This third-level categorization creates the deepest level of understanding on identifying technical debt as the nature of the debt is considered in the categorization.

2.2 *Related Work on Technical Debt Analysis on Enterprise-Level Solutions*

Although the technical debt concept has been extensively studied in the literature, there are few studies in relation to analysis of technical debt for enterprise software systems. Klinger et al. analyzed the technical debt by conducting interviews with technical architects related to enterprise-level solutions and made recommendations for organizations to manage technical debt with enterprise-level circumstances [9]. Another study in this area is on managing technical debt using an option-based approach for cloud-based solutions, where each option's ability to clear technical debt is analyzed [8]. There is also a study which focuses on the dependencies between client and vendor maintenance activities in enterprise-level software systems and empirically quantifying "the negative impact of technical debt on enterprise system reliability" [3]. The difference of our study is that we are also focusing on analyzing technical debt for enterprise software, but we base our analysis on *unbiased and ambiguous issue statements of project stakeholders* for a CRM platform.

2.3 *Salesforce.com*

Salesforce is a customer relationship management (CRM) solution that brings companies and customers together for improving marketing, sales, and services through connected products by understanding the needs and concerns of customers [16]. Salesforce is provided as a platform-as-a-service cloud solution. Being a cloud-based solution, the Salesforce platform offers basics of CRM solutions in terms of standard tools and processes, and also offers numerous ways of customization for its users, such as creation of custom objects and processes, as well as providing an online marketplace called AppExchange for Salesforce apps, components, and services [17].

Over 150,000 companies, across every industry, are supporting their business processes with Salesforce [18]. The increasing number of companies and therefore the subscribers are creating a large community of businesspeople, developers, administrators, and partners. In this community, independent software vendors (ISVs) build their solutions and publish them on AppExchange. Consulting partners build customizations based on the business requirements of their customers on this platform. Effective communication between all parties is very important in the software delivery life cycle of the Salesforce platform. The platform has its own

customizable processes, programming language, standard features, and technical jargon that facilitate customizations and collaboration across different stakeholders. These customization approaches make the Salesforce platform prone to technical debt.

3 Research Methodology

The research methodology employed in this study consists of six main phases: (i) review and selection of the technical debt categorization methods (summarized in Sect. 2); (ii) specification of the data repository used in this research which provides unbiased data regarding the issues encountered in software development (OrgConfessions); (iii) collecting issue data from OrgConfessions; (iv) categorization of the issues according to each technical debt categorization method; (v) validation of the categorization; and the (vi) analysis of the results.

Details of these phases are given below:

- (i) Specification and analysis stages of different technical debt categorization methods published in the literature were summarized in Sect. 2.
- (ii) We based our study on issue statements published as anonymous confessions by an independent software vendor (OrgConfessions). These issue statements are written by developers, administrators, consultants, and users working on the Salesforce platform.
- (iii) As of writing this paper, there are currently more than 700 entries published. The entries are not always written in-detail, and the confessors are usually using an informal voice. Moreover, they used the domain knowledge to express the issues succinctly. Hence, many of the confessions are ambiguous and cannot be categorized without knowing the context and the nature of the pertinent implementation. Therefore, the total number of entries analyzed in this study is lower than the total number of entries published, covering approximately the 40% of the total.
- (iv) Technical debt categorization was carried out by a certified Salesforce platform developer, having more than 2 years of experience in product development and consultancy in the platform.

The categorization process mainly included analysis of the issue definitions on OrgConfessions and determination of relevant technical debt categories based on three categorization approaches for each entry. These technical debt categorizations were briefly described in Sect. 2.

At the first level of detail, the entries are evaluated from the *intention* aspect to see whether the developers foresaw the consequences when they made an inappropriate decision (intentionally) or not (unintentionally). Example confessions for these categories are as follows:

- “People refuse to do changes anywhere but in production, since there “is no real test data” (confession #231) → *Intentional*
- “We installed a managed package that added a currency field onto every standard AND custom object.” (confession #129) → *Unintentional*

For the second level of categorization, the intentions of the decision-maker who eventually caused the debt were the criteria on evaluating the entries. These intentions are found by using the “technical debt quadrant” introduced by Fowler. Each category in the quadrant can be identified as *Reckless (R)*, *Prudent (P)*, *Inadvertent (I)*, and *Deliberate (D)*. The union of these categories constitutes the technical debt quadrant. The example entries falling in each quadrant section are as follows:

- “Users doing a Trailhead course and started installing apps in our prod org vs their own playground.” (confession #201) → *Reckless-Inadvertent*
- “35 users logging in with shared username/password that had System Admin access.” (confession #198) → *Reckless-Deliberate*
- “Validation rule to stop one spammer (hardcoded email address) from creating email-to-case.” (confession #349) → *Prudent-Deliberate*
- “Invited to new Chatter group called “ISV-ABC.” We all ignored it because it looked like a test. ABC actually stands for a territory: AMER – BUILD – CENTRAL.” (confession #307) → *Prudent-Inadvertent*

The categorization at the third level was performed by considering the software development process during which the debt injected. This evaluation is based on Alves et al.’s [10] 13 different categories. A sample set of confessions for these categories are as follows:

- “40 lookup fields on the Product object.” (confession #74) → *Design*
- “Request for a multi-select picklist with 98 values. When advised this was not the best practice and to rethink the need, they came back with a request for a picklist with 78 values.” (confession #84) → *People*
- “Sandbox not updated 10 years.” (confession #164) → *Process*
- “Multiple fields with same label.” (confession #147) → *Documentation*
- “Hard-coded user names in Apex classes.” (confession #39) → *Code*

The validation (v) and analysis phases (vi) are discussed in Sects. 4 and 5, respectively.

4 Validation Process

As mentioned above, one of the authors of this research, who works as a developer in a company developing new applications used in the Salesforce platform, has performed the technical debt categorization process. To validate the categorizations chosen by her, three different sets of 15 confessions were independently evaluated

Table 2 Matching entries for each level of categorization

Categorization level	Number of matching entries	Percentage of matching entries
1	41/45	91%
2	35/45	77%
3	34/45	75%

by five experts who also work at the same Salesforce consultant team with the author. Four of these experts work as software developers, whereas one of them works as a tester/quality assurance specialist. These experts have varying experiences in both software development and the Salesforce platform. Hence, in order to minimize the effect of validator experience on analysis, we employed cross validation. To this end, for example, the experts with less than 1 year of experience on the Salesforce platform but having nearly 10 years of experience in software development and those having around 2 years of experience both in Salesforce and software development evaluated the same set of confessions.

Before the evaluations, a short textual description for the technical debt approaches and categories was provided to the experts. Forty-five entries were chosen randomly from the set of categorized entries (confessions). This refers to validation of the 15% of the categorizations given by the researcher. As the confessions are vaguely stated, the experts' categorizations were not always consistent. For this reason, we employed the majority voting method in deciding on the correct technical debt categories for each categorization level. The categories found by the researcher were compared with the ones found by validators, and in the case of a tie, the researcher's categorization was assumed to be valid.

In the validation phase of this study, the focus was also on understanding whether different roles or experiences in software development or the Salesforce platform influenced the categorization process. Table 2 shows that most of the Salesforce experts were in consensus in categorization of the confessions (91% at Level 1, 77% at Level 2, and 75% at Level 3). However, due to ambiguities in the issue statements, and the existence of non-mutually exclusive categories (e.g., Design and Architecture) in technical debt categorization at Level 3, we concluded that the categorization levels can be revised to consider the ambiguous or unknown sourced debt issues.

As can be seen in Table 2 out of validated 45 entries, the percentage of matching entries are decreasing with increasing level of categorization. This shows that when the technical debt categories' granularity increases, the consistency of the decisions made in categorization decreases.

One reason for this decrease at Level 3 is that there is no clear distinction among the categories of this level. For example, in some cases, a debt categorized as "Design" could go under "Architecture" or "Code" or even "Requirement" categories as well.

Another cause of this decrease is the ambiguities in issue definitions. Since the actual process in the software development life cycle where the debt had been intro-

duced was unknown to researchers and the group of validators, making decisions at Level 3 was more difficult than the other two technical debt categorization levels.

5 Results and Discussion

The categorization of 300 confessions was performed as described in Sect. 3, and the results are summarized in Table 3.

Table 3 shows that, out of 300 entries analyzed, most of the issues are classified as *Intentional* at Level 1 and as *Reckless-Deliberate* at Level 2.

Mapping of the categories at Level 1 and Level 2 as shown in Table 4 revealed that most of the *Intentional* type technical debt corresponds to *Reckless-Deliberate* type debt according to Level 2 categorization, and most of the *Unintentional* type technical debt corresponds to the *Reckless-Inadvertent* type debt at Level 2.

As it can be seen in Table 3, *Design* and *People* type categories are the most frequent ones at Level 3. Hence, in Tables 5 and 6, we present a breakdown of the *Design* and *People* category confessions for Level 1 and Level 2. As shown in Table 5, most of the *Design* type technical debt are categorized as *Intentional* (113) and *Reckless-Deliberate* (103). On the other hand, most of the *People* type technical debt are categorized as *Intentional* (27) and *Reckless-Deliberate* (25) as

Table 3 Number of entries in each technical debt category

Level	Technical debt type	Number of entries
1	Intentional	219
	Unintentional	81
2	Reckless-Deliberate	202
	Reckless-Inadvertent	66
	Prudent-Deliberate	11
	Prudent-Inadvertent	21
3	Design	138
	People	51
	Process	41
	Documentation	24
	Code	16
	Test	13
	Requirement	5
	Infrastructure	5
	Service	4
	Architecture	3

Table 4 Mapping of Level 1 and Level 2 categories

		Level 2					
		R.D	R.I	PI	P.D		
Level 1	Intentional	200	6	2	11	219	
	Unintentional	2	60	19	0	81	

Table 5 Total number of entries in Level 1 vs Level 2 categorization when Level 3 category is Design

Detailed results for Level 3 = Design	Level 2					
	R.D	R.I	P.I	P.D		
Level 1	Intentional	103	3	2	5	113
	Unintentional	1	19	5	0	25

Table 6 Total number of entries in Level 1 vs Level 2 categorization when Level 3 category is People

Detailed results for Level 3 = People	Level 2					
	R.D	R.I	P.I	P.D		
Level 1	Intentional	25	1	0	1	27
	Unintentional	0	19	5	0	24

well. Additionally, the second frequent category for the *People* technical debt was *Unintentional* and *Reckless-Inadvertent*. Hence, we can say that most of the *People* type technical debt can also be classified as a *Reckless* technical debt.

When we analyzed the number of entries categorized at these three different levels, we observed that the first two levels of technical debt categorization are strongly correlated to each other. The most likely cause of this correlation is that the intention aspect is also covered by the Level 2 categorization defined by Fowler in the technical debt quadrant. Fowler states that “. . . the moment you realize what the design should have been, you also realize that you have an inadvertent debt,” pointing out that the correlation we mentioned between Level 1 and Level 2 categorizations is valid. By nature, the *Unintentional* debt cannot be known until the moment it is realized, and Fowler’s statement supports that. This also explains the same case for the *Inadvertent* debt category.

Below, we discuss the causes of the most frequently observed Level 3 technical debt types – the “Design,” “People,” and “Process” from a CRM perspective.

The “Design” type technical debt in the Salesforce platform can be attributed to requirements errors as well. It is not possible to distinguish design and requirement type errors due to ambiguity in issue definitions. Therefore, a design issue may also suggest an inefficiency in requirement elicitation and specification processes (such as requirements are not well defined or analyzed enough for a specific business need which leads to incorrect solutions in the Salesforce platform). Some of the design issues may be related to replicating an already existing third-party package or features in the Salesforce platform which ends up with a redundant development. Similarly, instead of using built-in components which already exist in the Salesforce platform, introducing new custom-made components or solutions that satisfy the same set of requirements is classified as “Design” debt.

We observed that the “People” type technical debt category is strongly related to faults caused by human errors and lack of user training. The technical debt introduced in this category is also linked to communication errors or lack of communication. People working on the Salesforce platform with different roles are

implicitly or explicitly mentioned in the confessions. The following are the entries for representing the different roles causing the “People” type technical debt at Level 3:

- “We can’t move to Lightning because our Dev Team refuses to learn JavaScript to write the Lightning components we need.” (confession #114) → The role of the person mentioned in this confession is Developer
- “10-year-old Org. Admin didn’t know how to customize nav bar. So they went in for each user and customized their tabs.” (confession #186) → The role of the person mentioned in this confession is Administrator
- “All managers insist on having a password that never expires.” (confession #216) → The role of the person mentioned in this confession is Businesspeople

We think that the technical debt linked to “Process” type is strongly related to the methodology followed in the development and delivery processes. Process-related debts are indicators of not following the software development best practices [19] accepted by the Salesforce community (i.e., not using the suggested deployment connections for deployment purposes).

In the light of these evaluations, answers to the research questions introduced in Sect. 1 are as follows:

R.Q.1: To what extent we can identify the causes of technical debt based on ambiguous issue statements?

We applied three different levels of categorization with varying degrees of granularity to evaluate technical debt in a CRM platform. The intentions, the carefulness, and the software development aspects involved in introducing debts have been identified based on ambiguous issue statements. The intentions and the carefulness of the technical debt introducers can be distinguished effectively with the use of the first two levels of categorization. When there is ambiguity in the issue definitions, high-level categories having lower granularity bring more precise results in the technical debt categorization process compared to more granular categorizations.

In more detailed categorizations, such as the Level 3 classification employed in this paper, determining the correct technical debt category becomes more challenging due to ambiguity in issue definitions. Moreover, there may also be issues corresponding to multiple technical debt categories. Therefore, a more granular classification would be needed to ensure flexibility in assigning multiple debt categories to same issue definitions.

R.Q.2: How do different technical debt categorization or identification methods help on analyzing the causes of technical debt in a customer relationship management (CRM) platform?

Three different categorization levels that we employed help tremendously in analyzing the debt. The first two levels summarize the intentions and the behavior in terms of carefulness of people and organizations, which is helpful in detecting the debt in terms of the company’s culture. The third level of categorization is helpful in the identification and the causes of technical debt in a more detailed

manner, since it gives a lower level understanding of the debt, in terms of where it was introduced in the first place. In order to make it more useful, we need more clear distinctions between categories, and we should consider multiple categories for each issue especially when we deal with ambiguous issue definitions.

In line with these answers, we made the following inferences that may be useful for further research on technical debt categorization in CRM products:

- When the ambiguity of the issue statements is high, multiple levels of categorization would be useful to analyze technical debt from different perspectives.
- It was understood that the intentions of the decision-makers in software delivery processes are very important in analyzing the technical debt incurred when delivering and maintaining software, especially for the debts introduced in the “Design” stage of the project/product.
- The most beneficial and clear technical debt categorization could be performed by the people who were involved in the customization and development of the Salesforce platform.
- Each technical debt may be related to one or more issues involving several different development activities or aspects. Hence, we may consider multiple categories for each issue rather than mapping each issue to just one category.

6 Conclusion

In this study, we investigated the technical debt in the Salesforce platform based on anonymous confessions from different Salesforce organizations by different consultants, developers, and administrators. These anonymous confessions were also ambiguous as the confessors failed to provide any kind of background information for the issues and referred to very specific implementation details in relation to the Salesforce platform. However, the value of using these ambiguous – but potentially unbiased – issue definitions is an important aspect of this study.

We investigated the effectiveness of different technical debt categorization approaches proposed in the literature. In particular, we defined three increasingly detailed levels of categorization. We observed that such a multi-granular approach may be useful to analyze the technical debt from different points of view.

According to our analyses, the first two levels of categorization are strongly correlated and can be unified in a single level. There is also a need to define each category in the CRM context for the third level of categorization which offers the higher granularity. It may also be possible to bring new levels of categories to cover different aspects of software development. Moreover, it may be necessary to customize categories to better support development in different domains.

References

1. P. Kruchten, R.L. Nord, İ. Ozkaya, Technical debt: From metaphor to theory and practice. *IEEE Softw.* **29**, 18–21 (2012)
2. N. Zazworka, R. Spinola, A. Vetro', A case study on effectively identifying technical debt, in *17th International Conference on Evaluation and Assessment in Software Engineering, Porto de Galinhas*, (2013)
3. N. Ramasubbu, C.F. Kemerer, Technical debt and the reliability of enterprise software systems: A competing risks analysis. *Manag. Sci.* **62**, 1487–1510 (2016)
4. Z. Codabux, B. Williams, G. Bradshaw and M. Cantor, An empirical assessment of technical debt practices in industry, *J. Softw. Evol. Process* **29**(10), e1894 (2017)
5. G. Skourletopoulos, R. Bahsoon, C. Mavromoustakis and G. Mastorakis, Predicting and quantifying the technical debt in cloud software engineering, (2014)
6. N. Zazworka, A. Vetro', C. Izurieta, Comparing four approaches for technical debt identification, *Softw. Qual. J.* **22**(3), 403–426 (2013)
7. G. Iuliia, Technical debt management in Russian software development companies, Master's Thesis, St. Petersburg University Graduate School of Management, (2017)
8. E. Alzaghoul, R. Bahsoon, CloudMTD: Using real options to manage technical debt in cloud-based service selection, in *4th International Workshop on Managing Technical Debt, MTD 2013*, (2013)
9. T. Klinger, P. Tarr, P. Wagstrom, C. Williams, An enterprise perspective on technical debt, (2011)
10. N. Alves, L. Ribeiro, V. Caires , T. Mendes, R. Spínola, Towards an ontology of terms on technical debt, in *6th IEEE International Workshop on Managing Technical Debt, Victoria*, (2014)
11. Salesforce, Customer Relationship Management (CRM), Salesforce, [Online]. Available: <https://www.salesforce.com/ap/definition/crm/>. Accessed 12 May 2020
12. G. Suryanarayana, G. Samarthyam, T. Sharma, *Refactoring for Software Design Smells: Managing Technical Debt*. Publisher: Morgan Kaufmann Publishers Inc., San Francisco, USA (Morgan Kaufmann, 2014)
13. S. Mcconnell, Construx, 1 January 2013. [Online]. Available: <https://www.construx.com/resources/whitepaper-managing-technical-debt/>. Accessed 5 Feb 2020
14. M. Fowler, martinfowler.com, 14 10 2009. [Online]. Available: <http://martinfowler.com/bliki/TechnicalDebtQuadrant.html>. Accessed 12 May 2020
15. W. Cunningham, The WyCash portfolio management system, (1992)
16. Salesis, What is salesforce? Salesforce, [Online]. Available: <https://www.salesforce.com/eu/products/what-is-salesforce/>. Accessed 12 May 2020
17. Salesforce, ISVforce guide, Salesforce, [Online]. Available: https://developer.salesforce.com/docs/atlas.en-us.packagingGuide.meta/packagingGuide/appexchange_intro.htm. Accessed 9 Feb 2020
18. Salesforce, Worlds number one CRM, Salesforce, [Online]. Available: <https://www.salesforce.com/campaign/worlds-number-one-CRM/>. Accessed 12 May 2020
19. Salesforce, Deploy enhancements from sandboxes, Salesforce, [Online]. Available: https://help.salesforce.com/articleView?id=changesets_about_connection.htm&type=5. Accessed 10 May 2020

Applying DevOps for Distributed Agile Development: A Case Study



Asif Qumer Gill and Devesh Maheshwari

1 Introduction

Agile approaches have fundamentally changed the way organizations develop and deploy software [1]. Agile approaches focus on iterative development and continuous improvement [2]. Agile development team aims to incrementally deliver the working software to operations team [3]. The operations team is then responsible for putting the software into the production environment. Despite the recent success with agile development, operations at large still work in isolation and slow compared to agile development teams [4]. Being agile in development, and not agile in operations, is one of the major concerns of agile teams. Lack of alignment and synchronization between the development and operations could lead to the problem of slow release and longer time to market of software solutions [5]. Isolated and slow operations in traditional settings could be collectively seen as a bottleneck in the overall value stream of software delivery [6]. In order to address this important concern, an alternative and integrated DevOps (development and operations) approach is emerging and getting vast attention from software-intensive organizations [7].

DevOps seems to be an interesting approach. However, its adoption in distributed agile software development is a challenging task [8–10]. This paper aims to address this important concern and presents a case study of DevOps adoption in distributed agile development environment. This study provides interesting insights and key success factors of DevOps adoption such as (1) small teams, (2) trust, (3) active communication and collaboration culture, (4) shared product vision and roadmap,

A. Q. Gill (✉) · D. Maheshwari
University of Technology Sydney, Ultimo, NSW, Australia
e-mail: asif.gill@uts.edu.au

(5) continuous feedback and learning culture, and (6) appreciation and excellent senior management support.

This chapter is organized as follows. Firstly, it describes the DevOps concepts. Secondly, it presents the DevOps case study results. Finally, it presents the discussion and conclusion.

2 DevOps

DevOps is defined as “a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production, while ensuring high quality” [6, 11]. The integrated DevOps brings together both the development and operations teams and seems to address the bottleneck of slow releases of software into production environment [12, 13]. The integration of DevOps is not a straightforward task and poses several technical and nontechnical challenges [14]. This paper presents a case study of DevOps adoption in an Australian software-intensive organization (ABC – coded name due to privacy concerns) for the distributed agile development and deployment of a real-time high-performance gaming platform. The experiences and learnings from this case study will help the readers to understand the DevOps process, its implementation, and key learnings.

3 DevOps Case Study

The ABC is an ASX listed entertainment organization. It offers gaming software solutions in Australia. Its vision is to create entertainment experiences where the passion, thrills, and enjoyment of the Australian way of life come alive. It is one of the world’s largest publicly listed gaming firms. It runs multiple gaming brands and has the ability to handle huge amount of daily real-time transactions (over a million) on a high-performance platform, and their digital capability allows them to deliver the same at the fast pace by using an agile approach. It has a flat organization structure, which is augmented by continuous feedback and learning culture.

3.1 Analytical Lens

ABC has been using DevOps in their distributed agile environment for more than 3 years. ABC DevOps case has been analyzed and reported by using the “Iteration” management capability layer (see Fig. 1) from the adaptive enterprise project management (AEPM) capability reference model [15]. The AEPM capability reference model specifies the services for adaptive or agile portfolio, program,

project, release, and iteration management layers. DevOps is one of the services embedded in the bottom “Iteration” layer of the APEM; hence, it has been deemed appropriate and used here as an analytical lens to systematically analyze the ABC DevOps case.

An iteration is a short time-boxed increment of a software. Iteration has embedded services, which are organized into three parts: pre-iteration, iteration implementation, and post-iteration implementation. Iteration team employs practice and tools to realize these services. Pre-iteration services include adaptive iteration planning, analysis, and architecture for the upcoming iteration. Iteration implementation refers to the integrated development and operations (DevOps) of current iteration in hand. It also involves automated testing and continuous deployment (CD) services [16]. The CD also includes continuous integration services (CI) [17]. The deployment covers the deployments in development, test, staging, and production environments. Code is design, which emerges as the DevOps progresses in small increments. However, design service can be used to document the technical design, if required. Heuristics refers to continuous learning or adaptation through iterative feedback and reviews.

3.2 Iteration Management

ABC is using an agile release cycle, which involves the development of prioritized product features in 2-week increments. Release cycle includes inception stage, which includes release planning, vision, and scope. Release cycle spans multiple iterations that frequently release working software increments into production. The ABC release cycle has been analyzed and detailed below using the iteration management capability layer items (see Fig. 1).

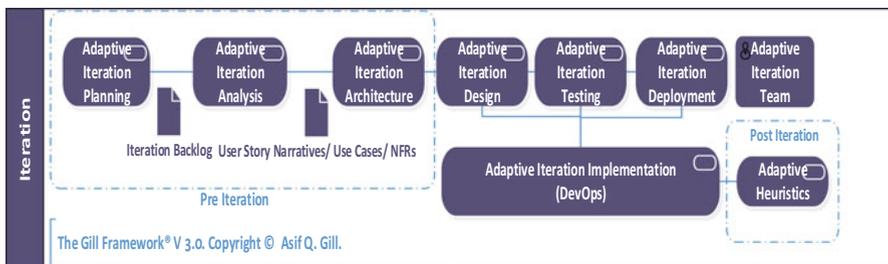


Fig. 1 APEM – showing iteration management layer. (Adapted from [15])

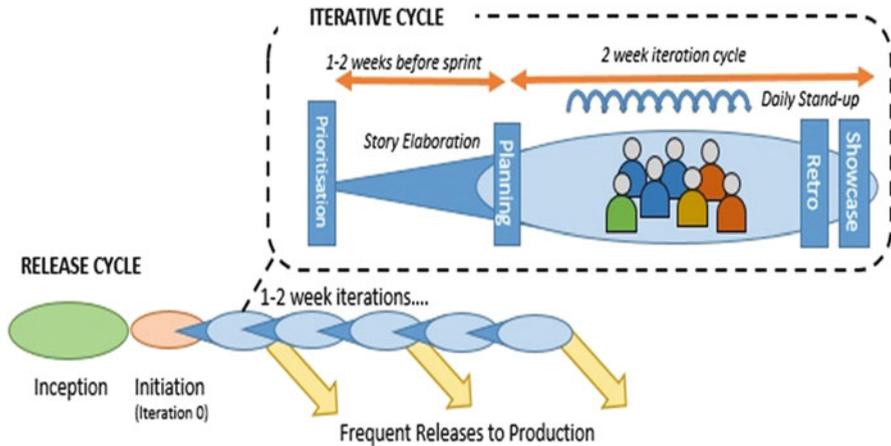


Fig. 2 APEM – ABC iterative cycle

Iteration Team

ABC has 70+ IT team members working on the gaming platform, which are organized into small geographically distributed feature teams. These teams are located across Sydney, Melbourne, and Brisbane. Each feature team size ranges from six to nine people. These teams are supported by three DevOps engineers. DevOps engineers create standard scripts, lay out foundation for execution, and guide teams to move in the right direction. It is important to note here that development teams take ownership to the larger extent to deliver features including DevOps tasks. DevOps engineers mainly facilitate the feature teams to smoothly deploy product increments into production environment. Feature teams continuously deploy code in test environment. However, the code is deployed into production twice in a week. Hence, teams delivering features take the ownership and responsibility of taking the code through to production and support it (Fig. 2).

Pre-Iteration

The iteration cycle of ABC has iteration 0, which is called the initiation stage. Iteration 0 is also a pre-iteration for next iteration (iteration 1). It means that iteration 1 planning, user stories, and architecture (story prioritization and elaboration) are detailed in iteration zero (0). Similarly, iteration 2 planning, user stories, and architecture are detailed in pre-iteration 1. This enables the team to have the user stories ready (analyzed, planned, and architected) before the start of next iteration.

Iteration Implementation (DevOps)

The bulk of the work is done during iteration implementation. Product increment user stories are implemented by the distributed agile teams using automated DevOps practices and tools. Development is done by using the Microservices Architecture style, in which application is decomposed into small independent fine-grained services in contrast to traditional coarse-grained service [18]. The application Microservices are deployed in the cloud (Amazon Web Services), which is also integrated with the ABC company infrastructure. Automated testing, functional and nonfunctional, is built into the DevOps process. The development team develops the code and automated tests, which are required to complete the user stories. Any new scenario identified by the team during the iteration implementation is also estimated and prioritized and, if required, is incorporated into the current or upcoming iterations. Code is a design. However, additional technical design, if required, is also done as a part of the user story implementation. The artifacts, other than the code, are captured on the source control wiki for information management and sharing.

Code is frequently checked into the version control system and is also peer reviewed. Once code has been peer reviewed and automated build on CI server is passed, it is merged into the mainline repository. Once the code is checked into the version control, the automated tests are run by the CI server again to verify that the change has not had any adverse impacts on the rest of the solution. This is to ensure the quality and integrity of the solution. There is a high level of automated test coverage. It is made sure that the relevant acceptance criteria have met and execution of the exploratory tests is done for any edge cases. Any identified issues are captured as comments in the story tracking tool for a given story and are fixed straight away by the person who developed the story, and then rechecked by the person verifying the story. It is the mindset of the team that all issues or defects have high priority and need to be fixed as early as possible. This is done to avoid the possibility of hanging over issues and technical debt.

In a nutshell, user stories cannot be deployed or considered “complete” until they are tested as a part of the automated test suite. User stories, acceptance tests, and defects are captured and tracked using the agile tool, which is called Mingle. The CD employs CI to ensure that the code base is always in a deployable state and that regression defects have not been inadvertently introduced. The CI is enabled using the “GO” CI integration server [19], which is responsible for deployment orchestration. It is also supported by the GitHub repository [20] for version control for both code and test scripts. Confluence (wiki) [21] is used for capturing supporting information that is not recorded in Mingle or GitHub. Ansible is used for preparing configuration [22]. Further, active communication and collaboration among distributed agile teams are enabled using the HipChat communication tool. Each user has their own login, every change is recorded showing who made the change, and for each check in, the associated Pivotal Tracker ID is referenced. Figure 3 summarizes the DevOps value stream.

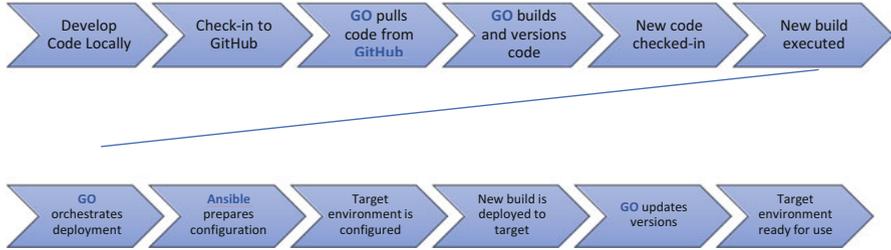


Fig. 3 DevOps value stream

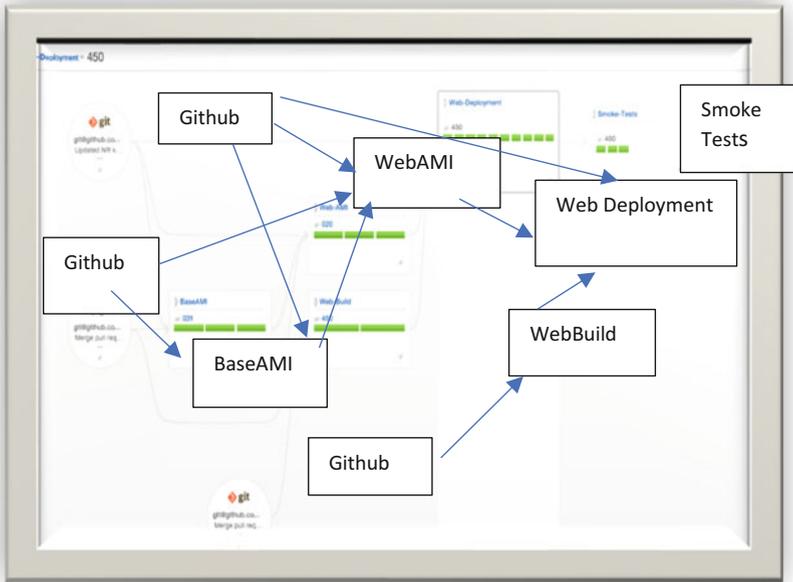


Fig. 4 Deployment pipeline

The CD of the overall DevOps process involves deployments in five different target environments: local development, shared development, testing, preproduction, and production environments. Local developments are done on the stand-alone machine or laptop. Shared development environment involves one or more components. Testing environment is for functional testing such as UAT. Preproduction is a production-like environment for performance testing and related bug fixing. Finally, production is a customer facing environment, which is duly monitored, operated, and supported by the DevOps team. Deployment pipeline can be traced from GitHub to Base AMI (Amazon Machine Instance) to Web AMI to Web Deployment (see example in Fig. 4).

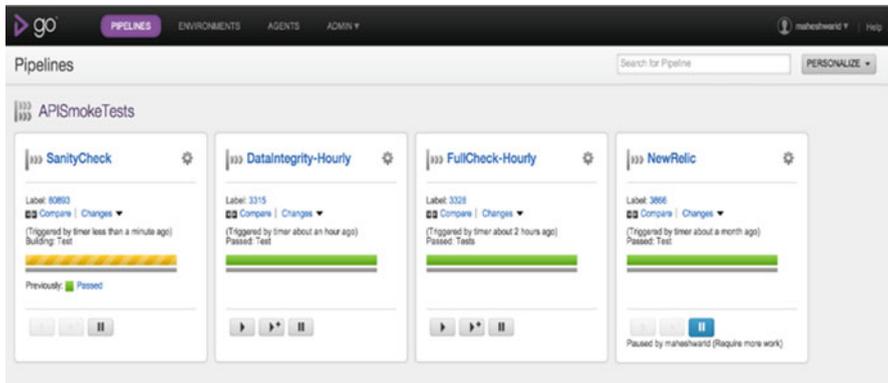


Fig. 5 Automated production checks

One of the goals of the DevOps is to make available the working solution into production environment as quickly as possible. However, the quality of the deployed code or solution is very important from target stakeholders’ perspectives. Therefore, ABC DevOps implements the additional automated production checks (see example in Fig. 5). It involves automated sanity test, which runs every few minutes and sends alert alarm on mobile to check any customer impact due to a deployment. These checks have been divided into 5 minutes and hourly checks based on their criticality and execution time.

Each iteration involves at least two showcases, one for technical understanding to internal team and one is for business external to customer. This enables the team to quickly identify and address any technical and business requirements-related concerns during the iteration. In addition to product owner, customer care team is also involved during the business showcase. Further, in order to keep the distributed agile feature teams aligned and synchronized, ABC maintains a card wall or portfolio of features (shared vision) and roadmap organized into the next 3, 6, 9, and 12 months. This helps the distributed feature teams to understand the holistic picture (shared vision and roadmap) while working on their local features.

Post-Iteration Implementation (Heuristics)

Post-iteration heuristics involves iteration retrospective. In addition to traditional retrospective, it also involves process self-assessment. The secondary process owners run regular self-assessments to ensure conformance to the mandates and records identified by the team. This is achieved by sighting the content in the nominated repositories against the self-assessment checklist. The quality manager, on a periodical basis, reviews the completed self-assessments and raise Improvement Tickets for any noncompliance issues that cannot be justified.

Table 1 DevOps case study analysis results summary

Iteration	Services	Practices	Tools	Key team roles
Pre-iteration	Planning Analysis Architecture	Planning Prioritization User story elaboration	Mingle Confluence (wiki) HipChat	Development team Iteration manager Product owner SME UXD
Iteration implementation	Design DevOps Testing Deployment	Technical design Automated testing CD CI Code peer review Change handling Technical showcase Business showcase	Mingle Confluence (wiki) GO GitHub Ansible HipChat	Development team DevOps engineer Iteration manager Product owner SME UXD Customer care team
Post-iteration implementation	Heuristics	Retrospective Improvement tickets	Mingle Self-assessment checklist	Development team DevOps engineer Iteration manager Quality manager Product owner SME UXD

The ABC organization’s DevOps case study analysis results summary is presented in Table 1. It is clear from the analysis that ABC has a well-established DevOps environment within the overall distributed agile development. We also learned that APEM reference model elements provided us with a structured mechanism or checklist to systematically analyze the DevOps case study and ensure that the important points are not overlooked.

4 Discussion and Conclusion

Setting up with smaller and trusted features teams to deliver features gave the ABC organization the flexibility to try out various mechanisms and technologies for delivering software. Active communication and collaboration culture and shared product vision and roadmap helped the ABC distributed agile teams to stay align and synchronized. Further, continuous feedback, learning, appreciation, and senior management support helped the teams to stay motivated to successfully implement the DevOps in their distributed agile environment over a period of 3 years. Microservices Architecture and DevOps are considered as a strong combination. However,

interestingly, the ABC digital delivery lead mentioned that “with growth we realized that we made lot of decisions like splitting monolithic application into multiple smaller services and created lot of micro services. On the one hand, we are seeing advantages of having micro services, but there is also a risk of having too many services which will in the future create more work of maintaining deployments, risk of having things implemented differently on each of the services, risk of having each services only serving few routes.” This seems to suggest that organizations should proceed with great caution when considering Microservices Architecture. Security could be an issue in a flexible DevOps environment. ABC deals with this issue through monthly security audit reviews on DevOps. ABC is currently looking at which fine-grained Microservices can be combined or consolidated into more coarse-grained traditional services. This chapter presented a DevOps implementation case study in a relatively different context of entertainment gaming industry. This case study provided us several insights which could be applied to other industrial contexts. It is clear from the case study analysis that DevOps is not all about technology; it is a mix of both technology and non-technology elements. DevOps is an emerging approach for digital innovation and transformation and marks the need for more empirical studies in this important area of practice and research.

References

1. A. Qumer, B. Henderson-Sellers. Construction of an agile software product-enhancement process by using an agile software solution framework (ASSF) and situational method engineering. In *31st Annual International Computer Software and Applications Conference (COMPSAC 2007)* (Vol. 1, pp. 539–542). IEEE (2007)
2. T. Dybå, T. Dingsøyr, What do we know about agile software development? *IEEE Softw.* **26**, 6–9 (2009)
3. A.Q. Gill, A. Loumish, I. Riyat, S. Han, DevOps for information management systems. *VINE J. Inf. Knowl. Manag. Sys.* **48**(1), 122–139 (2018)
4. M. Huttermann, *DevOps for Developers* (Apress, New York, 2012)
5. M. Virmani. *Understanding DevOps & bridging the gap from continuous integration to continuous delivery*, in *Fifth IEEE International conference on Innovative Computing Technology (INTECH 2015)*, (2015)
6. L. Bass, I. Weber, L. Zhu, *DevOps: A Software Architect’s Perspective* (Addison-Wesley, Westford, Massachusetts, 2015)
7. L.F. Wurster, R.J. Colville, J. Duggan. Market trends: DevOps — Not a market, but a tool-centric philosophy that supports a continuous delivery value chain. Gartner report, (2015). Available: <https://www.gartner.com/doc/2987231/market-trends-devops%2D%2Dmarket>
8. Y.I. Alzoubi, A.Q. Gill, A. Al-Ani, Distributed agile development communication: An agile architecture driven framework. *JSW* **10**(6), 681–694 (2015)
9. Y.I. Alzoubi, A.Q. Gill, An empirical investigation of geographically distributed agile development: The agile enterprise architecture is a communication enabler. *IEEE Access* **8**, 80269–80289 (2020). <https://doi.org/10.1109/ACCESS.2020.2990389>
10. G.B. Gbantous, A.Q. Gill, An agile-DevOps reference architecture for teaching enterprise agile. *Int. J. Learn. Teach. Educ. Res.* **18**(7), 128–144 (2019)

11. G. Bou Ghantous, A. Gill, DevOps: Concepts, practices, tools, benefits and challenges (2017). PACIS2017
12. G.B. Ghantous, and A.Q. Gill, DevOps reference architecture for multi-cloud IOT applications. In *2018 IEEE 20th Conference on Business Informatics (CBI)* (Vol. 1, pp. 158–167). IEEE (2018)
13. R. Macarthy, J. Bass. An empirical taxonomy of DevOps in practice. in *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, (2020)
14. M. de Bayser, L.G. Azevedo, R.F.G. Cerqueira, ResearchOps: The Case for DevOps in Scientific Applications. IFIP/IEEE IM 2015 Workshop: 10th International Workshop on Business-driven IT Management (BDIM), (2015)
15. A.Q. Gill, *Adaptive Cloud Enterprise Architecture* (World Scientific, Norske Telekom, Singapore, 2015)
16. S.J Humble, D. Farley, *Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation*, 1st edn. (Addison-Wesley Professional, Boston, 2010)
17. P.M. Duvall, S. Matyas, A. Glover, *Continuous Integration: Improving Software Quality and Reducing Risk*, 1st edn. (Addison-Wesley Professional, Boston, 2007)
18. S. Newman, *Building Microservices Designing Fine-Grained Systems* (O'Reilly Media, CA, 2015)
19. Go. Continuous delivery. Available: <http://www.go.cd/>. Access Date: 06 Apr 2020
20. GitHub. Where software is built. Available: <https://github.com/>. Access Date: 06 Apr 2020
21. Confluence. Available: <https://www.atlassian.com/software/confluence>. Access Date: 06 Apr 2020
22. Ansible. Available: <http://www.ansible.com/>. Access Date: 06 Apr 2020

Water Market for Jazan, Saudi Arabia



Fathe Jeribi , Sungchul Hong, and Ali Tahir 

1 Introduction

Jazan is a desert province located in the eastern side of the Red Sea [1, 25]. In Jazan, the average temperature is 30.1 degrees Celsius [2] and the annual average humidity is 66% [3]. This means humidity is relatively high in a desert area. The wind speed in Jazan is between 6.8 and 8 miles per hour (mph) annually, i.e., 3–3.57 meters per second (mps) [4].

This paper introduces a water market for the Jazan region. This market can be used to trade the amount of water between sellers and buyers. Water can be generated using water generator machines. Electricity for these machines can be supplied using solar energy and wind energy to reduce costs. The possibility of trading through the proposed market is experimented using math models and computer simulation.

1.1 Energy Source

Solar power can be defined as energy that is derived from sunlight for a variety of uses. And solar radiation can be converted into either electrical or thermal energy [5, 17]. Solar panels are devices that utilize sunlight to generate solar energy. The goal of solar energy technologies is to supply electricity, heating, and light for industries,

F. Jeribi (✉) · A. Tahir
Jazan University, Jazan, Saudi Arabia
e-mail: fjeribi@jazanu.edu.sa

S. Hong
Towson University, Towson, MD, USA

Table 1 Solar energy technologies

No.	Technology	Explanation
1	Photovoltaic systems	The goal of this technology is to generate electricity from sunlight directly
2	Solar hot water	The goal of this technology is to heat water using solar energy
3	Solar electricity	The goal of this technology is to generate electricity using the sun's heat
4	Passive solar heating and daylighting	The goal of this technology is to supply both light and heat for buildings using solar energy
5	Solar process space heating and cooling	In this technology, the sun's heat can be utilized for commercial and industrial purposes

businesses, and homes. Many technologies (Table 1) are utilized to benefit from solar energy [6]. In this paper, we assume that people use the SolarWorld SW 250 Poly as the solar panel device. In a good condition, this panel can produce 250 wh [7].

Wind energy can be defined as the process of collecting wind's kinetic energy through wind turbine. After that, it uses a generator to convert this energy into electrical energy [8, 15]. In terms of renewable energy, wind energy is considered one of fields that grows fast [9]. In terms of environmental benefits, wind energy has zero emissions [9]. Wind energy can be utilized for two goals: electricity generation and water pumping [9]. There are many benefits of wind energy. Some of them are minimizing the utilization of fossil fuels and decreasing imports of energy [10, 27]. We assume that people or businesses use 2.5–3 MW wind turbine. Onshore, the average electricity production of this wind turbine is 6 million kWh yearly. In average, this amount of electricity can support 1500 households [11].

An atmospheric water generator (AWG) is a machine that utilizes humidity with the goal of extracting water [12, 16]. It is considered as a substitute solution for gathering fresh water from air [13]. AVKO 365K is an example of AWG. This device can generate 1000 liters of water every day. In addition, it needs 8.2 kilowatts per hour to operate, or in other words, it needs 196.8 kwh per day [14].

2 Literature Review

Auction is a market that helps to sell the good by the bidding way. In this market, the seller has to decide the first price and the buyer provides the highest price [18], or the seller provides offers and the buyer provides bids [21]. In the auction market, the seller has to provide the lowest price that he or she will agree to receive, and the buyer has to provide the highest price that he or she agrees to pay. The trading can happen between sellers and buyers only if they agree on a price [23, 29]. There

are many examples of auction market. American Stock Exchange (AMEX) is one example of auction market [19].

The goals of auction are revenue maximization, information aggregation and revelation, valuation and price discovery, transparency and fairness, speed and low administrative tasks, and fostering competition [20]. The auction market could be retail or wholesale [18]. There are two classifications for auction based on the number of sellers and buyers, which are single and double. Single auction means that there are one seller and many buyers, and double auction means there are many sellers and many buyers [22, 28]. Electronic auction market can be defined as auction that occurs online [27].

3 Generation of Water from Desalination Plant and Air

Desalination plants use seawater to make water. This can cause local salinity problem. Water can be generated from air using an atmospheric water generator machine. People can utilize solar energy and wind energy to generate electricity, and doing this method, the watermaking facility does not need to be located near the seashore. This facility can be located even inland. This collected electricity can be utilized to operate water generator devices. These devices can help to make water through air and then stored in the water tank. The stored water can be sold to the water market. In addition, water generated from desalination plants can also be sold to the water market (Fig. 1). In this paper, the water market is developed for the Jazan region, Saudi Arabia. However, it can be adopted to any region or country easily.

4 Water Market Structure

Basically, this water market is a many-to-many type of market. In the water market, there are many sellers and buyers (Fig. 2). Sellers and buyers can trade together through the water market. A seller could be a small, medium, or large based on water volume. AVKO 365K model is an example of a small seller. The advantage of this model is that it could be located on hard-to-access area or transportation cost is high. Water well is an example of a medium seller. The price of it is low; however, it has limited volume and location. Desalination plant is an example of a large seller. It has low water prices; however, it has many disadvantages too. The process in the desalination plant is costly due to consumption of oil burning and electricity. Desalination plant has also an environmental problem, which is throwing the high-salinity water back into sea after completing desalinating the water. This process can result in the rise of percentage of salt in the sea locally and temporary, causing some damages to the marine life. In addition, desalination plant is expensive to build, and it is limited to a seashore. In the buyer side, house is an example of a small buyer.

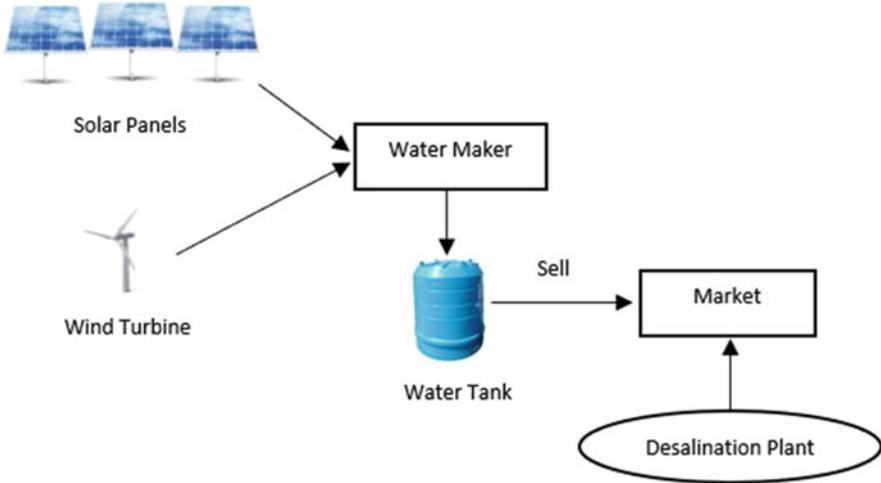
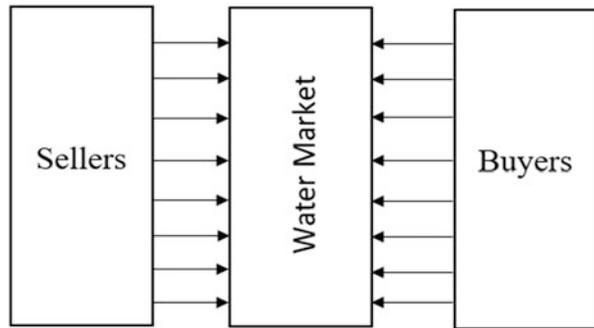


Fig. 1 Water from air structure

Fig. 2 Water market structure



Farm could be an example of a medium buyer. And factory is an example of a large buyer. The goal of this paper is to propose a water trading market for efficient water distribution through water trading.

5 Proposed Market Model

In this paper, there are one goal and two constraints. The goal is to find the maximum quantity trading of water, i.e., the maximum of multiplying water volumes and prices of a buyer (Eq. 1). The first constraint is that a seller's volume is greater than or equal to a buyer's volume (Eq. 2). The goal of this constraint is to make sure

that a seller has enough water volume for a buyer. The second constraint is that a seller’s price is less than or equal to a buyer’s price (Eq. 3).

$$\text{Max } \sum_i^i \sum_j^j (X_{.ij} \times V_j \times P_j) \tag{1}$$

$$V_i \geq V_j \tag{2}$$

$$P_i \leq P_j \tag{3}$$

$$X_{ij} = \begin{cases} 1: \text{ There is a matching} \\ 0: \text{ Otherwise} \end{cases} \tag{4}$$

where *i* is a seller and *j* is a buyer. *V_j* is a buyer volume and *P_j* is a buyer price. In Eq. 4, *X_{.ij}* has two values: 0 and 1, 1 meaning there is a matching between a seller and buyer and 0 meaning there is no matching between a seller and buyer. Figure 3 shows the process of the water market.

6 Analysis of Estimated Costs of Water Transportation

The total cost of water transportation is differentiated based on the distances of transportation, which are 30 km, 60 km, 120 km, and 240 km, for the convenience of the simulation. The factors that are used to calculate the total cost of water are original price, truck driver wage, truck depreciation cost, and fuel cost. Typically, one water truck can transport about 34,000 liters. Every 1000 liters costs \$1 as an original price without transportation costs. For example, the original cost of 34,000 liters is \$34.

To calculate truck driver wage, we assume that driver salary per year is \$73,000. In other words, the monthly payment will be \$6083.3, and the daily payment is \$202.8 (8 hours). To calculate truck depreciation cost, we assume that truck costs \$200,000 to purchase. In addition, the usual usage of a truck is 20 years. For 1 year, the cost of a truck will be 10,000 and for a day, the cost of a truck will be \$27.8. To calculate cost of fuel, we assume that the cost of truck fuel for 8 hours of driving is \$40. Based on the distances, Table 2 summarizes the anticipated costs of water transportation [24].

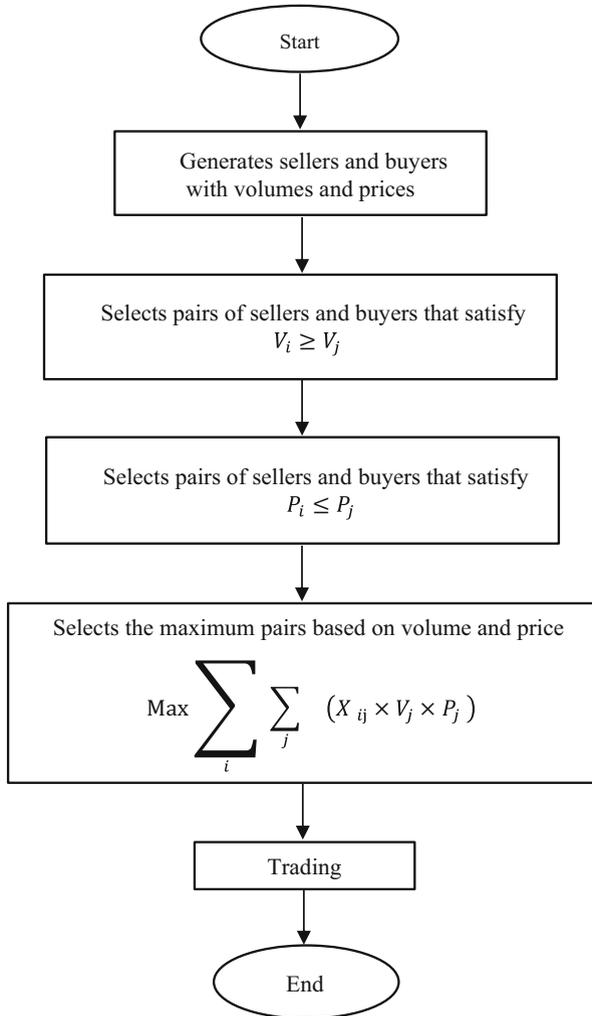


Fig. 3 The process of the water market

Table 2 The anticipated costs of water transportation

Destination in kilometers	30	60	120	240
Original price	\$34	\$34	\$34	\$34
Truck driver wage	\$25.35	\$50.7	\$101.4	\$202.8
Truck depreciation cost	\$3.48	\$6.95	\$13.9	\$27.8
Fuel cost	\$5	\$10	\$20	\$40
Total cost for one truck (34,000 liter)	\$67.83	\$101.65	\$169.2	\$304.6

7 Simulation System of the Water Market

A simulation model is designed based on the water price and volume. A uniform distribution of volume and price is also assumed. This water market does the following:

- Randomly, generates 30 volumes and 30 prices for sellers and buyers.
- Water volumes are between a quarter of water to three trucks of water. In other words, the water volumes are 8500, 17,000, 25,500, 34,000, 42,500, 51,000, 59,500, 68,000, 76,500, 85,000, 93,500, and 102,000 liters for the convenience of the simulation.
- Prices are calculated based on the sum of the original price, truck driver wage, truck depreciation cost, and fuel cost. In addition, it is based on profits. The range of profits is between 10% and 15%, which is selected randomly.
- Selects sellers' volumes that are greater than or equal to buyers' volumes.
- Selects sellers' prices that are less than or equal to buyers' prices.
- Selects the maximum of every seller matching. In other words, it will select the maximum of multiplying buyer volume and price.
- Shows the matching sellers and buyers. If there is no matching between a seller and a buyer, it will show that no trading will happen.

The different distances are reflected by the different costs of water.

7.1 Heuristic Algorithm

Even though the original formulation is an integer problem, it is difficult to solve an integer problem directly. Because of this difficulty, a heuristic algorithm is proposed. In this paper, heuristic algorithm does the following:

- First, it will select potential pairs of sellers and buyers based on volume.
- Second, from the selected pairs of sellers and buyers, pairs of sellers and buyers will be selected again based on the price.
- Third, among the selected pairs, it will select the maximum pairs based on multiplying volume and price. The algorithm will search the list from the smallest index sellers to the largest.

8 Simulation Results

The simulation result of a water market for the Jazan region is presented. This simulation result shows whether there are matchings between sellers and buyers.

Table 3 The volume and price for 30 sellers

Seller no.	Volume	Price
1	51000.0	51.0
2	8500.0	8.5
3	76500.0	76.5
4	34000.0	34.0
5	85000.0	85.0
6	102000.0	102.0
7	68000.0	68.0
8	102000.0	102.0
9	59500.0	59.5
10	93500.0	93.5
11	102000.0	102.0
12	17000.0	17.0
13	8500.0	8.5
14	34000.0	34.0
15	42500.0	42.5
16	25500.0	25.5
17	93500.0	93.5
18	76500.0	76.5
19	76500.0	76.5
20	8500.0	8.5
21	68000.0	68.0
22	17000.0	17.0
23	25500.0	25.5
24	68000.0	68.0
25	34000.0	34.0
26	34000.0	34.0
27	25500.0	25.5
28	85000.0	85.0
29	42500.0	42.5
30	68000.0	68.0

Tables 3, 4, and 5 show the result of single run of 30 sellers and buyers. Table 3 shows the volume and price for 30 sellers. Table 4 shows the volume and price for 30 buyers. Table 5 shows the result of all sellers and buyers with the result either trading or no trading.

Figure 4 summarizes the results of 40 simulations. It shows the number of matching cases and no matching cases for 40 simulations. The results show that the percentage of matching between sellers and buyers is more than no matching. This means that the water trading can help to find the matching between sellers and buyers. The simulation parameters are explained in Sect. 7.

Table 4 The volume and price for 30 buyers

Buyer no.	Volume	Price
1	76500.0	76.5
2	85000.0	85.0
3	42500.0	42.5
4	76500.0	76.5
5	59500.0	59.5
6	8500.0	8.5
7	25500.0	25.5
8	17000.0	17.0
9	85000.0	85.0
10	25500.0	25.5
11	85000.0	85.0
12	34000.0	34.0
13	34000.0	34.0
14	68000.0	68.0
15	93500.0	93.5
16	59500.0	59.5
17	59500.0	59.5
18	8500.0	8.5
19	51000.0	51.0
20	59500.0	59.5
21	51000.0	51.0
22	93500.0	93.5
23	102000.0	102.0
24	93500.0	93.5
25	8500.0	8.5
26	34000.0	34.0
27	102000.0	102.0
28	85000.0	85.0
29	34000.0	34.0
30	17000.0	17.0

Table 5 The result of the water trading market

Seller no. 1	Matching result
1	Can trade with a buyer 3
2	Can trade with a buyer 18
3	Cannot trade with any buyer
4	Can trade with a buyer 26
5	Can trade with a buyer 11
6	Can trade with a buyer 27
7	Can trade with a buyer 5
8	Can trade with a buyer 27
9	Can trade with a buyer 5
10	Can trade with a buyer 15
11	Can trade with a buyer 27
12	Can trade with a buyer 30
13	Can trade with a buyer 18
14	Can trade with a buyer 26
15	Can trade with a buyer 3
16	Can trade with a buyer 7
17	Can trade with a buyer 15
18	Can trade with a buyer 5
19	Can trade with a buyer 5
20	Can trade with a buyer 18
21	Can trade with a buyer 5
22	Can trade with a buyer 30
23	Can trade with a buyer 7
24	Can trade with a buyer 5
25	Can trade with a buyer 18
26	Can trade with a buyer 26
27	Can trade with a buyer 7
28	Can trade with a buyer 11
29	Can trade with a buyer 3
30	Can trade with a buyer 5

9 Conclusion

Water is a necessary resource for day-to-day activities of people or industries. In this paper, water trading market for the Jazan region is proposed. Generating water from air in this region could be available due the relatively high percentage of humid. For generating water, people or industries can use AWG as well as desalination. This generator can be operated by using solar and wind energy. Results of the simulation system showed that using water trading market can help sellers and buyers to trade water. And through trading, sellers and buyers can sell or buy easily and cost-effectively.

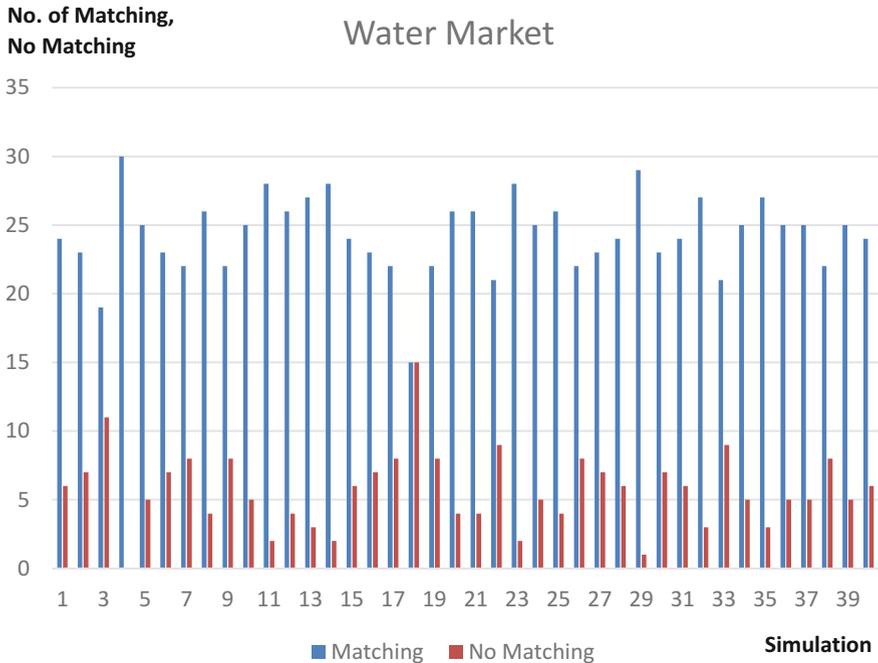


Fig. 4 The result of 40 simulations of the water market

References

1. A.M. Youssef, B. Pradhan, A.A. Sabtan, H.M. El-Harbi, Coupling of remote sensing data aided with field investigations for geological hazards assessment in Jazan area, Kingdom of Saudi Arabia. *Environ. Earth Sci.* **65**(1), 119–130 (2011). <https://doi.org/10.1007/s12665-011-1071-3>
2. Jazan region climate, <https://en.climate-data.org/asia/saudi-arabia/jazan-region-1998/>. Last accessed 2018/3/25
3. Average humidity in Jazan (Jazan Province), <https://weather-and-climate.com/average-monthly-Humidity-perc.j-z-n.Saudi-Arabia>. Last accessed 2018/4/5
4. Average weather in Jizan, Saudi Arabia, <https://weatherspark.com/y/102295/Average-Weather-in-Jizan-Saudi-Arabia-Year-Round>. Last accessed 2018/4/6
5. Solar energy, <https://www.seia.org/initiatives/about-solar-energy>. Last accessed 2018/4/8
6. Renewable energy, <https://www.renewableenergyworld.com/solar-energy/tech.html>. Last accessed 2018/4/10
7. SolarWorld SW 250 POLY PRO 250w solar panel, <https://www.solaris-shop.com/solarworld-sw-250-poly-pro-250w-solar-panel/>. Last accessed 2018/4/15
8. T. Agarwal, S. Verma, A. Gaurh, in Issues and challenges of wind energy, in *Proceedings of the 2016 International Proceedings on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pp. 67–72, <https://doi.org/10.1109/ICEEOT.2016.7754761>. IEEE, Chennai, India (2016)
9. A. Azad, M. Rasul, R. Islam, I.R. Shishir, Analysis of wind energy prospect for power generation by three Weibull distribution methods. *Energy Procedia* **75**, 722–727 (2015). <https://doi.org/10.1016/j.egypro.2015.07.499>

10. Wind power, <https://www.acciona.com/renewable-energy/wind-power/>. Last accessed 2018/5/2
11. Wind energy frequently asked questions (FAQ), <http://www.ewea.org/wind-energy-basics/faq/>. Last accessed 2018/5/5
12. Atmospheric water generation, <http://www.resotecwater.com/atmospheric-water-generators/>. Last accessed 2018/5/7
13. S. Suryaningsih, O. Nurhilal, Optimal design of an atmospheric water generator (AWG) based on thermo-electric cooler (TEC) for drought in rural area, in *Proceedings of AIP Conference* (2016). <https://doi.org/10.1063/1.4941874>
14. Akvo 36K, <http://akvosphere.com/akvo-atmospheric-water-generators/>. Last accessed 2018/6/3
15. A. Junyent-Ferre, Y. Pipelzadeh, T.C. Green, Blending HVDC-link energy storage and offshore wind turbine inertia for fast frequency response. *IEEE Trans. Sustain. Energy* **6**(3), 1059–1066 (2015). <https://doi.org/10.1109/tste.2014.2360147>
16. A. Tripathi, S. Tushar, S. Pal, S. Lodh, S. Tiwari, P.R.S. Desai, Atmospheric water generator. *Int. J. Enhanc. Res. Sci. Technol. Eng.* **5**(4), 69–72 (2016)
17. G.W. Crabtree, N.S. Lewis, Solar energy conversion. *Phys. Today* **60.3**, 37–42 (2007)
18. N. Prdić, B. Kuzman, The importance of auctions for agroindustrial products trade. *Ekonomika* **65**(1), 107–116 (2019). <https://doi.org/10.5937/ekonomika1901107p>
19. R. Aggarwal, J.J. Angel, The rise and fall of the Amex Emerging Company Marketplace. *J. Financ. Econ.* **52**(2), 257–289 (1999). [https://doi.org/10.1016/s0304-405x\(99\)00010-0](https://doi.org/10.1016/s0304-405x(99)00010-0)
20. P. Jehiel, B. Moldovanu, An economic perspective on auctions. *Econ. Policy* **18**(36), 269–308 (2003). <https://doi.org/10.1111/1468-0327.00107>
21. D. Friedman, J. Rust, The double auction market: institutions, theories, and evidence, in *Proceedings of the Workshop on Double Auction Markets*. Santa Fe, New Mexico (1993)
22. J. Trevathan, Electronic Auctions Literature Review (2005), <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.6.2353&rep=rep1&type=pdf>
23. What is an auction?, corporatefinanceinstitute.com/resources/knowledge/finance/auction/. Last accessed 2018/7/5
24. F. Jeribi, S. Hong, Viability of water making from air in Jazan, Saudi Arabia, in *Proceedings of the 19th Int'l Conference on Information & Knowledge Engineering*. Springer, Las Vegas, USA (2020). (Accepted but not published)
25. S.H. Mahmoud, A.A. Alazba, T. Amin, Identification of potential sites for groundwater recharge using a GIS-based decision support system in Jazan region-Saudi Arabia. *Water Resour. Manag.* **28**(10), 3319–3340 (2014). <https://doi.org/10.1007/s11269-014-0681-4>
26. A. Manyonge, R. Manyala, F. Onyango, J. Shichika, Mathematical modelling of wind turbine in a wind energy conversion system: Power coefficient analysis. *Appl. Math. Sci.* **6**, 4527–4536 (2012)
27. M.-Y. Lee, Y.-K. Kim, A. Fairhurst, Shopping value in online auctions: Their antecedents and outcomes. *J. Retail. Consum. Serv.* **16**(1), 75–82 (2009). <https://doi.org/10.1016/j.jretconser.2008.11.003>
28. P. Samimi, Y. Teimouri, M. Mukhtar, A combinatorial double auction resource allocation model in cloud computing. *Inform. Sci.* **357**, 201–216 (2016). <https://doi.org/10.1016/j.ins.2014.02.008>
29. T.N. Cason, Seller incentive properties of EPA's emission trading auction. *J. Environ. Econ. Manag.* **25**(2), 177–195 (1993). <https://doi.org/10.1006/jeem.1993.1041>

Modeling Unmanned Aircraft System Maintenance Using Agile Model-Based Systems Engineering



Justin R. Miller, Ryan D. L. Engle, Brent T. Langhals, Michael R. Grimaila, and Douglas D. Hodson

1 Introduction

Agile software development and model-based systems engineering (MBSE) are gaining momentum within the United States Air Force (USAF) and Department of Defense (DoD). However, the acquisition organizational culture has yet to abandon its reliance on evolutionary development approaches akin to the traditional waterfall methodology. Both of these methods seek to eliminate inefficiencies and to create a higher quality output that improves the flexibility of the system and increases transparency of the acquisitions process. The Air Force Chief Software Officer has also acknowledged agile software development as a “game-changing technology,” putting forth guidelines on implementing an agile framework [1]. Thus, the opportunity exists for academia to highlight when, where, and how these newer methods can be successful.

A unique challenge presented to the DoD and the Air Force is how to implement these new methods into already fielded systems when a capability gap is identified. The objective of this research is to explore agile-focused MBSE techniques to enable persistent and secure operations on unsecure or untrusted systems in a distributed environment. Systems that are fielded are of particular interest because of the reliance on the systems to perform their operational tasks that are often essential to the DoD mission. Rapid and effective improvement of these systems improves the overall effectiveness of DoD operations.

While typically performed in medium to large-scale teams, this research will use a small team, less than five personnel, of engineers to perform the MBSE and Agile development activities. The effectiveness of the small teams is another aspect of

J. R. Miller · R. D. L. Engle (✉) · B. T. Langhals · M. R. Grimaila · D. D. Hodson
Air Force Institute of Technology, Patterson, OH, USA
e-mail: ryan.engle@afit.edu

interest. Developing and adapting an agile process for a small team will present another challenge that will be tracked and documented throughout the research.

This research will focus on the maintenance operations of a commercially produced, large-scale unmanned aircraft system (UAS) that have an identified capability gap. The first step of this research is to create an architecture model of the existing system using the MBSE approach and tools. Agile practices will then be used to develop a product to fill the identified capability gap. Frequent iterations and transparency with the user will be essential to creating accurate models and producing an agile product to operate in this distributed environment.

The remainder of this paper is organized into three sections. The context of this project and key agile concepts will be outlined in the Background section. Next, the Methodology section will discuss the approach to developing and studying the iterative prototype development process. The last section will identify some anticipated outcomes of this research effort.

2 Background

The system under consideration is part of a distributed system. A distributed system is one that involves multiple computers rather than a single operating machine (centralized system) [2]. Distributed systems often use individual computers to execute activities but are contained on a cloud server that updates and stores information. They are designed to display the stored information to multiple machines or users at one time, appearing to be a single coherent system [3]. Distributed systems offer five main benefits: resource sharing, openness, concurrency, scalability, and fault tolerance [2]. However, the complexity of distributed systems makes them more difficult to design, implement, and test than centralized systems. Despite the benefits, there are key design issues that need to be considered: transparency, openness, scalability, security, quality of service, and failure management. While some of these are concerns for centralized systems, the complexity of distributed systems increases the planning and designing required.

To develop an effective solution, the system must first be comprehensively understood. Traditionally, a systems approach followed a static, document-centric process to describe system attributes and characteristics [4, 5]. Document-based approach focuses on generating documents that represent systems engineering artifacts such as concept of operations (ConOps) documents, requirements specifications, interface definition documents (IDDs), and system design specifications. However, these documents require constant upkeep and maintenance throughout the life cycle of the system to remain accurate and up to date. Additionally, this method does not emphasize developing a usable end product as the top priority.

MBSE is an alternative to the document-based approach used to describe and develop the system. Model-based systems engineering (MBSE) is a systems engineering approach that uses a system model. Such models are created using a modeling tool and language [6]. In contrast to the document-centric approach,

MBSE generates similar artifacts, but as a set of working or executable system models. These models contain elements describing the properties of the system. In MBSE, the diagrams and text artifacts are views of the system model [6]. When the model is updated, the changes are reflected in the views instantly. Using this approach drastically reduces the work required to update the diagrams and documents since the tool automatically propagates the changes throughout the model. As discussed previously, MBSE uses a modeling language to construct the system model; this research uses Systems Modeling Language (SysML) as the modeling language of choice, which is one of the most common modeling languages.

The Agile development model is an iterative approach to developing and delivering software [7]. The Agile approach contrasts with the traditional waterfall approach. Waterfall is a sequential development process focused on establishing requirements and design constraints before development and fully developing the system before deploying it to the customer [8]. The Agile development model uses incremental deployment to the user, delivering features and updating versions as the user gives feedback. This approach is especially useful in software models, where updating and upgrading, i.e., change, is more easily facilitated.

A key aspect of any development is the establishment of requirements and user desires. In Agile development, these take on the form of user stories. User stories are features that are short, descriptive sentences that highlight a desired functionality from the user's perspective. User stories are often written from the point of view of a user in the format of "As a *user*, I want to *activity* so that *business value*" [9]. These user stories provide direction and focus for which key features are desired. They also serve as a progress tracker. User stories can be tracked and used to determine which features have been implemented and which are yet to be completed.

Agile software development is based on four key values. The first value is "individuals and interactions over processes and tools" [7]. This indicates that Agile development is based on the people rather than set procedures. The next is "working software over comprehensive documentation" [7] which emphasizes the development of a minimally viable product (MVP) over documentation that slows development. The third key value is "customer collaboration over contract negotiation" [7] reinforcing the idea of customer involvement into the process. The last value of Agile software development is "responding to change over following a plan" [7], highlighting the flexibility required to properly use Agile development. These values, when implemented together, represent the basics of Agile software development [7, 9]. Although there are other Agile methods that use additional practices, the four key values will be the center of this research.

3 Methodology

The first portion of this research will use MBSE methods to fully depict the system of interest. In order to model the system, the researchers first need to understand

the process and the capability gap. However, Agile practices will still be used to maintain a streamlined development process and minimize the documentation needed. This is achieved by studying appropriate system documentation and interacting directly with the users of the system. Throughout the modeling process, an activity diagram will be created to document the maintenance process being improved. This activity diagram will describe the process and different decision points that will need to be reflected in the product to be produced. Updates will be made to the model as users give feedback in order to stay true to the actual process.

Using the key components outlined by the MBSE and Agile software development practices, an application will be developed to close the identified capability gap. User stories will be developed through interaction with the customer and through the use of the activity diagram and other MBSE artifacts. Throughout the development process, distributed system design issues will be addressed and discussed with the customers. Iterating and soliciting customer feedback frequently are keys to developing the product that the customer desires in an efficient manner. The first step is to create a graphical user interface (GUI) that is initially comfortable and understandable for the users. The backend software for the prototype will then be developed using Python. Each iteration for this portion of the process will involve the implementation of a list of desired features. The customer will then use the prototype and give feedback on the implemented features and features that they would like to see next. These iterations will continue until a full prototype is developed and sent to the users.

4 Preliminary Results and Anticipated Outcomes

Performing MBSE and Agile development as a small team will induce certain challenges but will also increase the efficiency of communication between the customer and the developers. The MBSE process will provide a critically needed understanding of the underlying system in an easily updatable format consistent with Agile practices, especially when compared to the document-centric counterpart. MBSE will also create artifacts that can be referenced as a baseline throughout the prototype development process in order to keep an understanding of the context of the process and issues that need to be addressed.

So far, some MBSE activities have been completed. When performing the MBSE activities, only the artifacts that add value to the project were created. The concept that Agile methodology needs to be flexible and be able to respond to user feedback is still used throughout the modeling process as well. The primary model used was an activity diagram. This allows the developers to understand the system process and identify the shortfalls that need to be addressed. Creating the activity diagrams also acted as a test to the developers' knowledge of the system that was discussed with the customers. The feedback given by the customers was used to refine the activity diagram, thus giving the developers a better understanding of the system as a whole.

Software development with a small team will increase the time required to deliver the full product but will ensure that confusion and miscommunication of requirements and features are kept to a minimum. While the distributed system will pose a challenge, active discussion with the user throughout the iterations will serve to answer many of the questions that arise during development. The overall product delivered will satisfy the user's needs and close the capability gap to improve operations.

Disclaimer The views expressed in this paper are those of the authors and do not reflect official policy or position of the US Air Force, the Department of Defense, or the US Government.

References

1. N.M. Chaillan, *Preferred Agile Framework* (Department of the Air Force, Washington DC, 2019)
2. I. Sommerville, *Software Engineering*, 10th edn. (Pearson Education Limited, London, 2019)
3. A. Tannenbaum, M. Van Steen, *Distributed Systems: Principles and Paradigms*, 2nd edn. (Prentice-Hall, Upper Saddle River, 2007)
4. Department of Defense Systems Management College, *Systems Engineering Fundamentals* (Defense Acquisition University Press, Fort Belvoir, 2001)
5. B.S. Blanchard, W.J. Fabrycky, *Systems Engineering and Analysis* (Pearson, Boston, 2011)
6. L. Delligatti, *SysML Distilled: A Brief Guide to the Systems Modeling Language* (Addison-Wesley, Upper Saddle River, 2014)
7. K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, J. Kern, B. Marick, R. C. Martin, S. Mellor, K. Schwaber, J. Sutherland, D. Thomas, Manifesto for agile software development, (2001). [Online]. Available: <https://agilemanifesto.org/>. Accessed 2020
8. W.W. Royce, Managing the development of large software systems: concepts and techniques, in *Proceedings of the 9th International Conference on Software Engineering*, (1987)
9. R. Knaster, D. Leffingwell, *SAFe 4.5 Distilled: Applying the Scaled Agile Framework for Lean Enterprises* (Addison-Wesley, Boston, 2019)

Benchmarking the Software Engineering Undergraduate Program Curriculum at Jordan University of Science and Technology with the IEEE Software Engineering Body of Knowledge (Software Engineering Knowledge Areas #1–5)



Moh'd A. Radaideh

1 Introduction

The Software Engineering Undergraduate Program at Jordan University of Science and Technology recently acquired and obtained an Accreditation from the Institute of Engineering and Technology (*hereinafter will be referred to as IET*) [3]. However, the Curriculum of the said Program needs further expansion to ensure its readiness for any potential ABET accreditation in the future as well as its readiness for training programs, professional licensing, and certification of specialties in the Software Engineering (SWE). The SWEBOK-V3.0 introduced 15 SWE-KAs. Some of them are not fairly covered or addressed in the said Curriculum. Table 1 lists these 15 SWE-KAs. Table 2 lists the SWE Courses of the SWE-Curriculum at JUST [1].

This paper is meant to elaborate on the coverage shortages of the first 5 of the 15 SWE-KAs across the various courses of the SWE Program Curriculum at JUST, while the coverage of the remaining ten SWE-KAs will be addressed in two separate papers (P#2 and P#3) that will follow this paper. These first 5 of the 15 SWE-KAs are as follows:

1. **SWE-KA#1: Software Requirements.** The Software Requirements are the needs and the constraints that must be met by a software system. The Software Requirement KA is concerned with the whole software requirement process including the elicitation, analysis, specification, validation, and management of the software requirements during life cycle of the software product. Chapter 1 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to

Md. A.Radaideh (✉)

Department of Software Engineering, Jordan University of Science and Technology, Irbid, Jordan
e-mail: Maradaideh@just.edu.jo

Table 1 SWE-KAs (Software Engineering Body of Knowledge – SWEBOK-V3.0)

P#1		P#2		P#3	
SWE-KA#	SWE knowledge areas (SWEBOK-V3.0) [2]	SWE-KA#	SWE knowledge areas (SWEBOK-V3.0) [2]	SWE-KA#	SWE knowledge areas (SWEBOK-V3.0) [2]
1	Software Requirements	6	Software Configuration Management	11	SWE Professional Practice
2	Software Design	7	SWE Management	12	SWE Economics
3	Software Construction	8	SWE Process	13	Computing Foundation
4	Software Testing	9	SWE Models and Methods	14	SWE Math. Foundation
5	Software Maintenance	10	Software Quality	15	Engineering Foundation

Table 2 The SWE courses of the SWE-Curriculum at JUST

SWE courses at JUST (SWE-Curriculum) [1]					
SE210	Java Programming [42]	SE321	Software Requirements Eng. [47]	SE430	Software Testing [50]
SE220	Software Modelling [43]	SE323	Software Documentation [48]	SE431	Software Security [51]
SE230	Fund. of Software Engineering II [44]	SE324	Software Architecture and Design [49]	SE432	Software Engineering for Web Applications [52]
SE310	Visual Programming [45]	SE326	Software Engineering Lab 1 [56]	SE440	Project Management [53]
SE320	Systems Analysis and Design [46]	SE471	Client/Server Programming [55]	CS318	Human-Computer Interaction (<i>Elective</i>) [54]
SE441	Software Quality Assurance [57]				

the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [7–13].

2. **SWE-KA#2: Software Design.** The Software Design is the process of specifying the internal structure of the software system to fulfill the requirements. Such a process defines the architecture of the software system in terms of its components and the interfaces between these components. During this process, software engineers produce various models describing various points of view of the system. Chapter 2 of the SWEBOK-V3.0 elaborates on this SWE-KA.

However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [14–23].

3. **SWE-KA#3: Software Construction.** The Software Construction KA concerns about the creation of the software system through a combination of coding, verification, testing, integration, and debugging. Chapter 3 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [24–27].
4. **SWE-KA#4: Software Testing.** Software Testing is the process of measuring the produced output matches the expected output. This process tries to reveal faults in the systems that may produce unexpected results or making the system vulnerable to various security attacks. Chapter 4 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [14, 28–32].
5. **SWE-KA#5: Software Maintenance.** Over time, the system evolves during new requirements or changes in the environment. The Software Maintenance ensures the operations of the software system after its delivery to the customer. Chapter 5 of the SWEBOK-V3.0 elaborates on this SWE-KA. However, readers can refer to the many references that are listed at the end of the said chapter. Examples of these references are listed as well at the end of this paper [33–38].

Section 3 provides the details of our research approach (e.g., *in the three Parts of this research*) involves the reflection of the various topics of these SWE-KAs onto the various courses of the SWE-Curriculum at JUST. It is worth mentioning that although this paper measures the coverage of the said SWE-Curriculum at JUST with the SWE-KAs, our innovative approach is general and can be applied to other SWE academic programs.

The findings of the first part (P#1) of this research gave a decent degree of compliance in the cases of the SWE-KAs of *Software Requirements* [7–13], *Software Design* [14–23], and *Software Testing* [14, 28–32], while the compliance is partial in the cases of the SWE-KAs of *Software Construction* [24–27] and *Software Maintenance* [36–41].

This paper is organized in several sections. Section 1 is this Introduction one. Section 2 discusses the related work. Section 3 elaborates on the innovative research approach followed in carrying out this research work. Section 4 composes five subsections such that each of them elaborates on the coverage of one of the first five SWE-KAs in the SWE Program Curriculum of JUST. Section 5 summarizes the findings of this paper, and a set of recommendations are made for possible enhancements on the said SWE Program Curriculum at JUST to make it more compliant with the first 5 of the 15 SWE-KAs and thus to improve its readiness for any potential ABET Accreditation in the future. Section 6 concludes this paper, and last but not the least, the References of this paper are listed.

2 Related Work

Early efforts toward organizing the teaching of SWE include, but not limited to, a paper by Bernhart, M., et.al. “Dimensions of Software Engineering Course Design” [4], and another one by Shaw, M. “Software Engineering Education: A Roadmap” [5]. Nevertheless, it is very important that software engineers read through *The Mythical Man-Month* book of Brooks FP [6].

Garousi et al. [39] conducted a literature review to study the collaboration between the software industry and SWE from an academic perspective to bridge the gap between these two large communities. They found that collaboration between the two is rather poor. To overcome this problem, they listed various challenges that would stand in the ways of any such collaboration, and they recommended best practices to be followed to surmount this obstacle. Similar to the effort in [39], to bridge the gap between academia and industry, this research studies the coverage of a SWE program with the SWE-KA and hence makes the program ready for provisional licensing, accreditation, and training paradigms.

Meziane et al. [40] compared between computer science and SWE programs in English universities in light of the knowledge areas in each field. They concluded that there are indeed many differences between the CS and the SWE curricula in England.

Fox et al. [41] from UC Berkeley shared their experiences in teaching SWE as Massive Open Online Courses (MOOCs) and Small Private Online Courses (SPOCs) to develop a SWE-Curriculum. They highlighted six interesting challenges in SWE education: (1) students do not have enough time to study the materials given; (2) the lack of an industrial expertise on the part of the SWE faculty; (3) there are so many SWE methodologies and it’s hard to choose and focus on any one of them; (4) the lack of good practical SWE textbooks; (5) it is very expensive for educational institutions to host and deploy the ever-increasing number of tools that support various SWE methodologies; and finally, the main challenge comprising (6) industry always complains about the quality of SWE education. To address these challenges, the authors of [41] argued that well-designed MOOCs and SPOCs courses improve the quality of SWE offerings due to the benefit of having a large community with whom to discuss issues and challenges.

Similar to the abovementioned related works, this research attempts to improve the quality of SWE graduates by improving the SWE-Curriculum itself. However, this research is significantly different as it evaluates the compliance of the SWE-Curriculum at JUST with the SWE-KAs (e.g., *this paper is concerned with the SWE-KAs#1–5, while the following two papers, P#2 and P#3, will address the remaining SWE-KAs*) in terms of the content coverage. Also, it measures the coverage of the main topics of the SWE-KAs in the expected learning outcomes of the SWE-Curriculum. To the best of the author’s knowledge, this research is the first to measure the coverage of the SWE-KAs in any SWE-Curriculum in terms of its contents as well as the learning outcomes of the courses in the SWE-Curriculum.

3 Research Methodology

The research approach followed to carry out this research work can be outlined in the following steps:

1. Dividing the SWE Knowledge Areas into the following two groups:
 - (a) SWE *Specialization-Related* Knowledge Areas (e.g., SWE-KAs#1–10)
 - (b) SWE *Professional-Support-Related* Knowledge Areas (e.g., SWE-KAs#10–15).
2. Splitting (e.g., *due to the size of this research work*) the *Specialization-Related* group (e.g., SWE-KAs#1–10) into two parts such that *P#1* covers the first five SWE-KAs (e.g., SWE-KAs#1–5) and *P#2* covers the second five SWE-KAs (e.g., SWE-KAs#6–10). Consequently, *P#3* of this research covers the *Professional-Support-Related* group of SWE Knowledge Areas (e.g., SWE-KAs#11–15).
3. Inspecting the coverage of the SWE-KAs (e.g., *for each of P#1, P#2, and P#3*) in the SWE Program Curriculum of JUST.
 - (a) The coverage of the *Specialization-Related* Knowledge Areas (e.g., *P#1* and *P#2*) will be inspected across the SWE Specialization course work across the said SWE-Curriculum.
 - (b) The coverage of the *Professional-Support-Related* Knowledge Areas will be inspected across the university with the college-required courses across the said SWE-Curriculum.
 - (c) The syllabus of each course in the said SWE-Curriculum will be carefully reviewed to figure out its coverage of the various topics of the various SWE-KAs.
 - (d) The latest version of the said SWE-Curriculum (e.g., *the IET Accredited SWE Program Curriculum*) is used for this research work.
4. Classifying the coverage of each SWE-KA (e.g., *for each of P#1, P#2, and P#3*) in the said SWE-Curriculum into one of the following levels:
 - (a) *Fully Compliant* (100%). This indicates that the concerned SWE-KA is fully covered across one or more of the courses of the said SWE-Curriculum.
 - (b) *Highly Compliant* (75–<100%). This indicates that the concerned SWE-KA is highly covered across one or more of the courses of the said SWE-Curriculum.
 - (c) *Partially Compliant* (50–<75%). This indicates that the concerned SWE-KA is partially covered across one or more of the courses of the said SWE-Curriculum.
 - (d) *Poorly Compliant* (<50%). This indicates that the concerned SWE-KA is poorly covered across one or more of the courses of the said SWE-Curriculum.

5. Classifying the coverage of the main topics of each SWE-KA (e.g., for each of *P#1*, *#2*, and *#3*) in the course learning outcomes (CLOs) of the said SWE-Curriculum. The learning outcomes are obtained from the syllabi of the courses of the said SWE-Curriculum, which can be accessed at [1]. The CLO coverage of each SWE-KA is classified into one of the following levels:
 - (a) *Fully Compliant* (100%). This indicates that all main topics of the concerned SWE-KA are fully declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
 - (b) *Highly Compliant* (75–<100%). This indicates that most of the main topics of the concerned SWE-KA are declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
 - (c) *Partially Compliant* (50–<75%). This indicates that part of the main topics of the concerned SWE-KA are declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
 - (d) *Poorly Compliant* (<50%). This indicates that few of the main topics of the concerned SWE-KA are declared as course learning outcomes (CLOs) across one or more of the courses of the said SWE-Curriculum.
6. Shortage identification and making recommendations. At the end of each part (e.g., *P#1*, *P#2*, and *P#3*), the coverage compliances (or shortages) will be identified and recommendations will be made such that new courses shall be introduced into the curriculum or existing ones shall be enhanced and/or revised in the said SWE-Curriculum.
7. Verifying the achievement of the research prime objective. The overall purpose of this work is to facilitate the potential ABET Accreditation of the SWE Undergraduate Program of JUST.

4 SWE-KAs Coverage in the SWE-Curriculum at JUST

The following set of tables (Tables 3, 4, 5, 6, and 7) illustrate the coverage of each of the first five SWE-KAs in the various SWE courses at JUST:n

4.1 Coverage of the SWEKA#1 (Software Requirements)

Table 3 concludes the following:

1. The SWE-KA#1 (Software Requirements) seems to be *fully covered* (100%) in the SWE Program Curriculum at JUST through the following courses: (i) SE230 Fundamental of Software Engineering, (ii) SE321 Software Engineering Requirements, and (iii) SE430 Software Testing.

Table 3 SWE-KA#1 (Software Requirements) and its Coverage in the SWE-Curriculum at JUST

Software Requirements Knowledge Area (SWBOK-V3.0)	Covered in the following SWE courses	Declared in the following CLOs
1. Software Requirements Fundamentals	SE321/SE230	CLO1 and CLO4 of SE321 [47] CLO2 of SE230 [44]
1.1. Definition of a Software Requirement	SE321/SE230	
1.2. Product and Process Requirements	SE321/SE230	
1.3. Functional and Non-functional Requirements	SE321/SE230	
1.4. Emergent Properties	SE321/SE230	
1.5. System Requirements and Software Requirements	SE321/SE230	
1.6. Quantifiable Requirements	SE321/SE230	
2. Requirements Process	SE321/SE230	CLO1 of SE321
2.1. Process Models	SE321/SE230	
2.2. Process Actors	SE321/SE230	
2.3. Process Support and Management	SE321/SE230	
2.4. Process Quality and Improvement	SE321/SE230	
3. Requirements Elicitation	SE321	CLO3 of SE321
3.1. Requirements Sources	SE321	
3.2. Elicitation Techniques	SE321	
4. Requirements Analysis	SE321	CLO2 of SE321
4.1. Requirements Classification	SE321	
4.2. Conceptual Modelling	SE321	
4.3. Architectural Design and Requirements Allocation	SE321/SE230	
4.4. Requirements Negotiation	SE321	
4.5. Formal Analysis	SE321	
5. Requirements Specification	SE321	CLO4 of SE321
5.1. System Definition Document	SE321	
5.2. System Requirements Specification	SE440	
5.3. Software Requirements Specification	SE440	

(continued)

Table 3 (continued)

Software Requirements Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in the following CLOs
6. Requirements Validation	SE321	CLO7 and CLO8 and CLO9 of SE321
6.1. Requirements Reviews	SE440/SE441	
6.2. Prototyping	SE440/SE441	
6.3. Model Validation	SE440/SE441	
6.4. Acceptance Tests	SE440/SE441	
7. Practical Considerations	SE321/SE430	CLO6 of SE321
7.1. Iterative Nature of the Requirements Process	SE321	
7.2. Change Management	SE321	
7.3. Requirements Attributes	SE321	
7.4. Requirements Tracing	SE321	
7.5. Measuring Requirements	SE321	
8. Software Requirements Tools	SE321	CLO9 of SE321

Table 4 SWE-KA#2 (Software Design) and its Coverage in the SWE-Curriculum at JUST

Software Design Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
1. Software Design Fundamentals	SE324	CLO 4 of SE230CLO1 of SE324
1.1. General Design Concepts	SE324/SE230/SE220	
1.2. Context of Software Design	SE324	
1.3. Software Design Process	SE324	
1.4. Software Design Principles	SE324	
2. Key Issues in Software Design	SE324	CLO6 of SE210CLO5 of SE310CLO1 of SE324
2.1. Concurrency	SE324/SE432	
2.2. Control and Handling of Events	SE324	
2.3. Data Persistence	SE324	
2.4. Distribution of Components	SE324	
2.5. Error and Exception Handling and Fault Tolerance	SE324/SE310/SE210	
2.6. Interaction and Presentation	SE324	
2.7. Security	SE324	
3. Software Structure and Architecture	SE324	CLO2 of SE324
3.1. Architectural Structures and Viewpoints	SE324	
3.2. Architectural Styles	SE324	
3.3. Design Patterns	SE324	
3.4. Architecture Design Decisions	SE324	
3.5. Families of Programs and Frameworks	SE324	
4. User Interface Design	CS318	CLO ** of CS318
4.1. General User Interface Design Principles	CS318	
4.2. User Interface Design Issues	CS318	
4.3. The Design of User Interaction Modalities	CS318	
4.4. The Design of Information Presentation	CS318	
4.5. User Interface Design Process	CS318	

(continued)

Table 4 (continued)

Software Design Knowledge Area (SWEBOOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
4.6. Localization and Internationalization	CS3/8	
4.7. Metaphors and Conceptual Models	CS3/8	
5. Software Design Quality Analysis and Evaluation	SE324	CLO2 of SE324
5.1. Quality Attributes	SE324	
5.2. Quality Analysis and Evaluation Techniques	SE324	
5.3. Measures	SE324	
6. Software Design Notations	SE324	CLO3 of SE324
6.1. Structural Descriptions (Static View)	SE324	
6.2. Behavioural Descriptions (Dynamic View)	SE324	
7. Software Design Strategies and Methods	SE324	CLO3 of SE324
7.1. General Strategies	SE324	
7.2. Function-Oriented (Structured) Design	SE324	
7.3. Object-Oriented Design	SE324	
7.4. Data Structure-Centered Design	SE324	
7.5. Component-Based Design (CBD)	SE324	
7.6. Other Methods	SE324	
8. Software Design Tools	SE324	CLO3 of SE324

Table 5 SWE-KA#3 (Software Construction) and its coverage in the SWE-Curriculum at JUST

Software Construction Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
1. Software Construction Fundamentals		
1.1. Minimizing Complexity	SE440	CLO4 of SE440
1.2. Anticipating Change	SE440	
1.3. Constructing for Verification		
1.4. Reuse	SE440	
1.5. Standards in Construction	SE440/SE441	
2. Managing Construction		
2.1. Construction in Life Cycle Models		CLO1 of SE210 CLO1 of SE310 CLO1 and CLO2 of SE430
2.2. Construction Planning		
2.3. Construction Measurement		
3. Practical Considerations		
	SE430/SE321/SE230	
3.1. Construction Design		
3.2. Construction Languages		
3.3. Coding	SE210/SE310	
3.4. Construction Testing	SE441/SE430	
3.5. Construction for Reuse		
3.6. Construction with Reuse		
3.7. Construction Quality		
3.8. Integration		
4. Construction Technologies		
4.1. API Design and Use	SE 310	CLO3 of SE210 CLO4 and CLO5 of SE310

(continued)

Table 5 (continued)

Software Construction Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
4.2. Object-Oriented Runtime Issues		
4.3. Parameterization and Generics	SE326	
4.4. Assertions, Design by Contract, and Defensive Programming		
4.5. Error Handling, Exception Handling, and Fault Tolerance for Distributed Software	SE310/SE210/SE324/SE430	
4.6. Executable Models		
4.7. State-Based and Table-Driven Construction Techniques		
4.8. Runtime Configuration and Internationalization		
4.9. Grammar-Based Input Processing		
4.10. Concurrency Primitives	SE371/SE324	
4.11. Middleware		
4.12. Construction Methods		
4.13. Constructing Heterogeneous Systems	SE371	
4.14. Performance Analysis and Tuning		
4.15. Platform Standards		
4.16. Test-First Programming		
5. Software Construction Tools		CLO3 of SE310
5.1. Development Environments	SE210/SE310/SE326	
5.2. GUI Builders	SE310/SE326	
5.3. Unit Testing Tools	SE430	
5.4. Profiling, Performance Analysis, and Slicing Tools		

Table 6 SWE-KA#4 (Software Testing) and its coverage in the SWE-Curriculum at JUST

Software Testing Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
1. Software Testing Fundamentals	SE430/SE230/SE441	CLO5 of SE230CLO1 of SE430CLO5 of SE230
1.1. Testing-Related Terminology	SE430/SE230	
1.1.1. Definitions of Testing and Related Terminology	SE430/SE230	
1.1.2. Faults vs. Failures	SE430	
1.2. Relationship of Testing to Other Activities	SE430/SE441	
2. Test Levels	SE430	CLO1 and CLO2 of SE430CLO ** of CS318
2.1. The Target of the Test	SE430/SE230	
2.1.1. Unit Testing	SE430/SE441	
2.1.2. Integration Testing	SE430/SE441	
2.1.3. System Testing	SE430/SE441	
2.2. Objectives of Testing	SE430/SE441	
2.2.1. Acceptance/ Qualification Testing	SE430/SE441	
2.2.2. Installation Testing	SE430/SE441	
2.2.3. Alpha and Beta Testing	SE430/SE441	
2.2.4. Reliability Achievement and Evaluation	SE430/SE441	
2.2.5. Regression Testing	SE430	
2.2.6. Performance Testing	SE430	
2.2.7. Security Testing	SE430/SE441	
2.2.8. Stress Testing	SE430/SE441	
2.2.9. Back-to-Back Testing		
2.2.10. Recovery Testing	SE430	
2.2.11. Interface Testing	SE430	

(continued)

Table 6 (continued)

Software Testing Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
2.2.12. Configuration Testing	SE430/SE441	
2.2.13. Usability and Human Computer Interaction Testing	SE430/CS318	
3. Test Techniques	SE430/SE441	CLO2 and CLO3 of SE430
3.1. Based on the Software Engineer's Intuition and Experience	SE430	
3.2. Input Domain-Based Techniques	SE430	
3.3. Code-Based Techniques	SE430	
3.4. Fault-Based Techniques	SE430	
3.5. Usage-Based Techniques	SE430	
3.6. Model-Based Testing Techniques	SE430	
3.7. Techniques Based on the Nature of the Application	SE430	
3.8. Selecting and Combining Techniques	SE430	
4. Test-Related Measures	SE430	CLO4 of SE430
4.1. Evaluation of the Program Under Test	SE430	
4.2. Evaluation of the Tests Performed	SE430	
5. Test Process	SE430	CLO6 of SE430
5.1. Practical Considerations	SE430	
5.2. Test Activities	SE430	
6. Software Testing Tools	SE430/SE441	CLO5 of SE430
6.1. Testing Tool Support	SE430	
6.2. Categories of Tools	SE430	

Table 7 SWE-KA#5 (Software Maintenance) and its coverage in the SWE-Curriculum at JUST

Software Maintenance Knowledge Area (SWEBOK-V3.0)	Covered in the following SWE courses	Declared in a CLO
1. Software Maintenance Fundamentals	SE230/SE441	
1.1. Definitions and Terminology	SE230/SE441	
1.2. Nature of Maintenance	SE230/SE441	
1.3. Need for Maintenance	SE230/SE441	
1.4. Majority of Maintenance Costs	SE230	
1.5. Evolution of Software		
1.6. Categories of Maintenance		
2. Key Issues in Software Maintenance	SE230/SE441	
2.1. Technical Issues		
2.2. Management Issues		
2.3. Maintenance Cost Estimation		
2.4. Software Maintenance Measurement		
2.5. Maintenance Processes		
2.6. Maintenance Activities		
3. Techniques for Maintenance		
3.1. Program Comprehension		
3.2. Reengineering		
3.3. Reverse Engineering		
3.4. Migration		
3.5. Retirement		
4. Software Maintenance Tools		

2. All main topics of the SWE-KA#1 (Software Requirements) are *fully declared as learning outcomes (100%)* in the SWE Program Curriculum at JUST through the following courses: (i) SE230 Fundamental of Software Engineering, (ii) SE321 Software Engineering Requirements, and (iii) SE430 Software Testing.

4.2 Coverage of the SWE-KA#2 (Software Design)

Table 4 concludes the following:

1. The SWE-KA#2 (Software Design) seems to be *fully covered (100%)* in the SWE Program Curriculum at JUST through the following courses: (i) SE210 Java Programming, (ii) SE220 Software Modelling, (iii) SE230 Fundamental of Software Engineering, (iv) SE310 Visual Programming, (v) SE324 Software Architecture and Design, (vi) SE432 Software Engineering for Web Applications, and (vii) CS318 Human-Computer Interaction.
2. All main topics of the SWE-KA#2 (Software Design) are *fully declared as learning outcomes (100%)* in the SWE Program Curriculum at JUST through the following courses: (i) SE210 Java Programming, (ii) SE230 Fundamental of Software Engineering, (iii) SE310 Visual Programming, (iv) SE324 Software Architecture and Design, (v) SE432 Software Engineering for Web Applications, and (vi) CS318 Human-Computer Interaction.

4.3 Coverage of the SWE-KA#3 (Software Construction)

Table 5 concludes the following:

1. The SWE-KA#3 (Software Construction) seems to be *partially covered (50–75%)* in the SWE Program Curriculum at JUST through the following courses: (i) SE210 Java Programming, (ii) SE326 SWE Lab 1, (iii) SE230 Fundamentals of SWE, (iv) SE310 C# Visual Programming, (v) SE321 SWE Requirements, (vi) SE324 Software Architecture and Design, (vii) SE430 Software Testing, (viii) SE440 Software Project Management, (ix) SE441 Software Quality Assurance, (x) and SE371 Client-Server.
2. All main topics of the Knowledge Area 31 (Software Construction) are *partially declared as learning outcomes (50–75%)* in the SWE Program Curriculum at JUST through the following courses: (i) SE210 Java Programming, (ii) SE310 Visual Programming, (iii) SE430 Software Testing, and (iv) SE440 Software Project Management.

4.4 Coverage of the SWE-KA#4 (Software Testing)

Table 6 concludes the following:

1. The SWE-KA#4 (Software Testing) seems to be fully covered (100%) in the SWE Program Curriculum at JUST through the following courses: (i) SE230 Fundamental of Software Engineering, (ii) SE430 Software Testing, and (iii) SE441 Software Quality Assurance.
2. All main topics of the SWE-KA#4 (Software Testing) are fully declared as learning outcomes (100%) in the SWE Program Curriculum at JUST through the following courses: (i) SE230 Fundamental of Software Engineering, (ii) SE430 Software Testing, and (iii) CS318 Human-Computer Interaction.

4.5 Coverage of the SWE-KA#5 (Software Maintenance)

Table 7 concludes the following:

1. The SWE-KA#5 (Software Maintenance) seems to be partially covered (50–75%) in the SWE Program Curriculum at JUST through the following courses: (i) SE230 Fundamental of Software Engineering and (ii) SE441 Software Quality Assurance.
2. All main topics of the SWE-KA#5 (Software Maintenance) are not declared as learning outcomes (0%) in the SWE Program Curriculum at JUST.

5 Discussion and Recommendations

This paper evaluated the compliance of the Software Engineering Undergraduate Program Curriculum with the SWE-KAs#1–5 of the SWEBOK-V3.0 of the IEEE Computer Society. Table 8 provides an overall view of the coverage of these five SWE-KAs in the SWE-Curriculum at JUST.

According to Table 8, the said compliance is either Fully Compliant (100%) or Partially Compliant (50–75%). In addition to measuring the content coverage of the SWE-KAs in the said SWE-Curriculum, this research also measured the explicit indication of the main topics of the SWE KSs as course learning outcomes (CLOs) in the said SWE-Curriculum. To that end, the CLO coverage of the first four SWE-KAs is consistent with the content coverage. However, all main topics of the last SWE-KA (Software Maintenance) are not indicated as learning outcomes by any course of the said SWE-Curriculum.

According to Table 9, to ensure full compliance of the Software Engineering Program Curriculum at JUST with the SWE-KAs#1–5, the SWE Program at JUST

Table 8 AN overall view of the coverage of the SWE-KAs#1-5 in the SWE-Curriculum at JUST

SWE-KA	Coverage in the SWE-Curriculum at JUST															
	Coverage	SE210	SE220	SE326	SE230	SE310	SE320	SE321	SE324	SE323	SE430	SE431	SE432	SE440	SE441	SE371
Software Requirements	<i>Fully Compliant</i>				✓		✓				✓					
Software Design	<i>Fully Compliant</i>	✓			✓	✓		✓	✓			✓				
Software Construction	<i>Partially Compliant</i>	✓		✓	✓		✓	✓	✓		✓		✓	✓		✓
Software Testing	<i>Fully Compliant</i>				✓						✓			✓		
Software Maintenance	<i>Partially Compliant</i>				✓				✓					✓		

Table 9 Compliance of the SWE-Curriculum at just with the SWE-KAs#1-5

SWE-KA	SWE-Curriculum compliance with the SWE-KAs#1-5			
	Fully Compliant (100%)	Highly Compliant (75-100%)	Partially Compliant (50-75%)	Poorly Compliant (below 50%)
Software Requirements	✓			
Software Design	✓			
Software Construction			✓	
Software Testing	✓			
Software Maintenance			✓	✓ (CLO Coverage)

shall address any Partially Compliant or Poorly Compliant issues in the said SWE-Curriculum. To achieve that, this paper recommends the following:

1. Adding a new course on *Software Construction*. Such course is strongly recommended to be based on Chap. 3 of the SWEBOK-V3.0 to ensure that all required Software Construction-related topics are covered.
2. Adding a new course on *Software Maintenance*. Such course is strongly recommended to be based on Chap. 5 of the SWEBOK-V3.0 to ensure that all required Software Maintenance-related topics are covered.

6 Conclusions

This paper reflects the compliance of the Software Engineering Undergraduate Program at JUST with the first 5 of the 15 SWE KAs presented in the Software Engineering Body of Knowledge of the IEEE Computer Society.

Three SWE-KAs (*SWE-KA#1 Software Requirements*, *SWE-KA#2 Software Design*, and *SWE-KA#4 Software Design*) of these five SWE-KAs were found to be *fully covered* in the said SWE Program Curriculum, while the other two (e.g., *SWE-KA#3 Software Construction* and *SWE-KA#5 Software Maintenance*) were found to be partially covered. Therefore, it was recommended in the previous section to introduce two new courses (e.g., *Software Construction* and *Software Maintenance Courses*) into the SWE-Curriculum at JUST.

The remaining SWE-KAs will be addressed in the following parts of this research (e.g., P#1 and P#2) that will be completed in the seen future.

Acknowledgments The author would like to thank Dr. Ahmed Shatnawi for his valuable input and careful proofreading of the final version of this paper. Also, he would like to thank Dr. M. Smadi and Dr. M. Hammad for their input to the early draft of this paper.

References

1. The Curriculum of the Software Engineering Undergraduate Program at Jordan University of Science and Technology, http://www.just.edu.jo/FacultiesandDepartments/it/Departments/SE/SiteAssets/Pages/Programs/IET-SE_English_StudyPlan2016.pdf
2. P. Bourque, R. Dupuis, Guide to the software engineering body of knowledge. IEEE Computer Society, Los Alamitos (2004) - SWEBOK-V3.0, <https://www.computer.org/education/bodies-of-knowledge/software-engineering>
3. Institute of Engineering and Technology Site – IET Accreditation, <https://www.theiet.org/career/accreditation/academic-accreditation/>
4. M. Bernhart, T. Grechenig, J. Hetzl, W. Zuser, Dimensions of software engineering course design, ICSE 2006, Shanghai, China, May 20–28 (2006), pp. 667–672
5. M. Shaw, Software engineering education: A roadmap, ICSE - Future of SE Track, (2000), pp. 371–380
6. F.P. Brooks, *The Mythical Man-Month*, Anniversary edn. (Addison-Wesley, Boston, 1975)

7. I. Sommerville, *Software Engineering*, 9th edn. (Addison-Wesley, 2011)
8. K.E. Wiegers, *Software Requirements*, 2nd edn. (Microsoft Press, 2003)
9. I. Alexander, L. Beus-Deukic, *Discovering Requirements: How to Specify Products and Services* (Wiley, 2009)
10. C. Potts, K. Takahashi, A.I. Antón, Inquiry-based requirements analysis. *IEEE Softw.* **11**(2), 21–32 (1994)
11. A. van Lamsweerde, *Requirements Engineering: From System Goals to UML Models to Software Specifications* (Wiley, 2009)
12. O. Gotel, C.W. Finkelstein, An analysis of the requirements traceability problem, in *Proc. 1st Int'l Conf. Requirements Eng., IEEE*, (1994)
13. N.A. Maiden, C. Ncube, Acquiring COTS software selection requirements. *IEEE Softw.* **15**(2), 46–56 (1998)
14. ISO/IEC/IEEE 24765:2010 Systems and Software Engineering—Vocabulary, ISO/IEC/IEEE, (2010)
15. IEEE Std. 12207–2008 (a.k.a. ISO/IEC12207:2008) Standard for systems and software engineering—software life cycle processes, IEEE, (2008)
16. IEEE Std. 1069–2009 Standard for information technology—systems design—software design descriptions, IEEE, (2009)
17. ISO/IEC 42010:2011 Systems and software engineering—recommended practice for architectural description of software-intensive systems, ISO/IEC, (2011)
18. L. Bass, P. Clements, R. Kazman, *Software Architecture in Practice*, 3rd edn. (Addison-Wesley Professional, 2013)
19. J.H. Allen, et al., *Software Security Engineering: A Guide for Project Managers* (Addison-Wesley, 2008)
20. T. DeMarco, *The Paradox of Software Architecture and Design*, (Stevens Prize Lecture, 1999)
21. D. Budgen, *Software Design*, 2nd edn. (Addison-Wesley, 2003)
22. I. Jacobson, G. Booch, J. Rumbaugh, *The Unified Software Development Process* (Addison-Wesley Professional, 1999)
23. G. Booch, J. Rumbaugh, I. Jacobson, *The Unified Modeling Language User Guide* (Addison-Wesley, 1999)
24. S. McConnell, *Code Complete*, 2nd edn. (Microsoft Press, 2004)
25. S.J. Mellor, M.J. Balcer, *Executable UML: A Foundation for Model-Driven Architecture*, 1st edn, (Addison-Wesley, 2002)
26. L. Null, J. Lobur, *The Essentials of Computer Organization and Architecture*, 2nd edn. (Jones and Bartlett Publishers, 2006)
27. A. Silberschatz, P.B. Galvin, G. Gagne, *Operating System Concepts*, 8th edn. (Wiley, 2008)
28. S. Naik, P. Tripathy, *Software Testing and Quality Assurance: Theory and Practice* (Wiley-Spektrum, 2008)
29. M.R. Lyu, *Handbook of Software Reliability Engineering* (McGraw-Hill and IEEE Computer Society Press, 1996)
30. H. Zhu, P.A.V. Hall, J.H.R. May, Software unit test coverage and adequacy. *ACM Comput. Surv.* **29**(4), 366–427 (1997)
31. S. Yoo, M. Harman, Regression testing minimization, selection and prioritization: A survey. *Softw. Test. Verification Reliab.* **22**(2), 67–120 (2012)
32. S.H. Kan, *Metrics and Models in Software Quality Engineering*, 2nd edn. (Addison-Wesley, 2002)
33. IEEE Std. 14764–2006 (a.k.a. ISO/IEC14764:2006) Standard for software engineering—software life cycle processes—maintenance, IEEE, (2006)
34. P. Grubb, A. Takang, *Software Maintenance: Concepts and Practice*, 2nd edn. (World Scientific Publishing, 2003)
35. H.M. Sneed, Offering software maintenance as an offshore service, in *Proc. IEEE Int'l Conf. Software Maintenance (ICSM 08), IEEE*, (2008), pp. 1–5
36. J.W. Moore, *The Road Map to Software Engineering: A Standards-Based Guide* (Wiley-IEEE Computer Society Press, 2006)

37. A. April, A. Abran, *Software Maintenance Management: Evaluation and Continuous Improvement* (Wiley-IEEE Computer Society Press, 2008)
38. M. Kajko-Mattsson, Towards a business maintenance model, in *Proc. Int'l Conf. Software Maintenance, IEEE*, (2001), pp. 500–509
39. V. Garousi, K. Petersen, B. Ozkan, Challenges and best practices in industry-academia collaborations in software engineering: A systematic literature review. *Inf. Softw. Technol.* **79**, 106–127 (2016)
40. F. Meziane, S. Vadera, A comparison of computer science and software engineering programmes in English universities, in *17th Conference on Software Engineering Education and Training*, (2004). Proceedings, pp. 65–70. IEEE
41. A. Fox, D.A. Patterson, R. Ilson, S. Joseph, K. Walcott-Justice, R. Williams, Software engineering curriculum technology transfer: lessons learned from MOOCs and SPOCs (2014). UC Berkeley EECS Technical Report
42. SE210 Java programming, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE210.pdf>
43. SE220 Software modelling, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE220.pdf>
44. SE230 Fundamentals of software engineering, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE230.pdf>
45. SE310 Visual programming, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE310.pdf>
46. SE320 System analysis and design, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE320.pdf>
47. SE321 Software requirements engineering, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE321.pdf>
48. SE323 Software documentation; <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE323.pdf>
49. SE324 Software architecture & design, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE324.pdf>
50. SE430 Software testing, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE430.pdf>
51. SE431 Software security, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE431.pdf>
52. SE432 Software engineering for web applications, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE432.pdf>
53. SE440 Project management, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE440.pdf>
54. CS318 Human-computer interaction, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/CS318.pdf>
55. SE471: Client server programming, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE471.pdf>
56. SE326: Software engineering lab, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE326.pdf>
57. SE441: Software quality assurance, <http://www.just.edu.jo/~ahmedshatnawi/syllabus/SE441.pdf>

Moh'd A. Radaideh is a Senior Member of the IEEE, the IEEE Computer Society, and the IEEE Education Society. He received his BENG & MENG degrees in Electrical and Computer Engineering from Yarmouk University and Jordan University of Science and Technology, consequently in 1987 & 1989, and his Ph.D. degree in Electrical and Computer Engineering (Software Engineering) from McMaster University (Canada) in 2000. He is currently an Associate Professor with the Department of Software Engineering, Jordan University of Science & Tech. **Profile:** http://www.just.edu.jo/admissionuploads/staff_cv/maradaideh.pdf.

A Study of Third-Party Software Compliance and the Associated Cybersecurity Risks



Rashel Dibi, Brandon Gilchrist, Kristen Hodge, Annicia Woods, Samuel Olatunbosun, and Taiwo Ajani

1 Introduction

Third-party software (TPS) are a great investment for companies. They help companies better manage their day-to-day processes and, in some cases, create a better user interface for their customers to use. The idea behind acquiring third-party cloud management tools is to offset what native tools cannot manage or do not see [1]. Customers expect certain capabilities and accessibility to a company and its' products; however, companies may not have the time or ability to support such customer needs especially with continuing evolution of the Internet and technology.

1.1 *Third-Party Software in the Cloud*

Most purchased TPS are in the cloud. The biggest benefit of third-party tools is visibility into a distributed cloud environment [1]. This is a plus and one of the main reasons why companies decide to purchase cloud-based TPS. Cloud-residing data provides a measure of stability to clients, especially in situations where natural and man-made disasters are present risks. Having a TPS also makes it easier on the administrator because they do not have to focus on things such as systems upgrades

R. Dibi · B. Gilchrist · K. Hodge · A. Woods · S. Olatunbosun (✉)
Department of Computer Science, Norfolk State University, Norfolk, VA, USA
e-mail: r.s.dibi@spartans.nsu.edu; bjgilchrist@nsu.edu; k.hodge102989@spartans.nsu.edu;
a.woods77749@spartans.nsu.edu; sbolatunbosun@nsu.edu

T. Ajani
Department of Computer Information Systems, Ferrum College, Ferrum, VA, USA
e-mail: tajani@ferrum.edu

and other maintenance issues. This leaves ample time to focus on other needs and issues at the company.

General Compliance Issues and Cybersecurity Risk

When it comes to compliance management, the ability to maintain and protect information, remediate problems, and provide adequate compliance reports is essential [2]. Companies have generally overlooked Information Technology Compliance and cybersecurity until recent years. In 2019, the total number of reported third-party breaches was 368. This number increased from 328 in 2018. In addition, the number of records exposed in these breaches skyrocketed to 273% last year, from just over 1.7 billion in 2018 to 4.8 billion in 2019 [3].

Problem Motivation and Importance

- (i) **Problem:** TPS companies are a continuous cybersecurity threat. Because of this, their data and the companies that they provide a service to are in jeopardy of being hacked. Moreover, regarding compliance, companies do not always have the correct measures to ensure that they are legally protected when hacking occurs.
- (ii) **Motivation:** Hackers are always looking for new ways and new products to retrieve data from. Companies with TPS provide more incentive to hackers.
- (iii) **Importance:** Cybersecurity and compliance protocols are needed for TPS. Providers may have companies' data from several countries within their network. No matter what kind of service that the company provides, any given company that they work with are liable to have sensitive personal information that could potentially end up in the wrong hands.

2 Literature

Security is an integral part of functional socioeconomic systems especially considering that people want their personal information protected from hackers and breaches. Security professionals are working diligently to ensure the safety of all people, especially during this time of the coronavirus pandemic when hacking is at the highest rate. The status of security risks is at a peak during the pandemic because of increased online activities and traffic. People are doing a fairly good job of selecting the TPS, but third parties need to be reviewed on a more regular basis to make sure that they keep up with standards, company requirements, as well as local and federal statutes [4]. When one thinks about security and the many attending risks, completing annual checks with the security can improve overall security functions in the security systems. Additionally, ensuring that the TPS are

adhering to the guidelines that are set out in the contract is just as important. It is not just about preventing breaches, but also making sure that the proper protocols are set in place when it happens. Risks are inevitable when addressing cybersecurity for TPS, but how one prepares for potential threats and work to improve the company computer systems can impact the chances of hackers gaining access to the system. Hackers work tirelessly to hack into systems, and there is no guarantee that they will be successfully apprehended. The number of fugitives residing in the United States is difficult to pinpoint because arrest warrants may be issued for minor offenses, such as a failure to appear for a traffic violation, or for more serious matters, such as when criminal suspects are on the run [5]. This information shows how easily criminals can get away with committing crimes such as hacking systems online specifically. One study estimates that two million criminal warrants may be active at any time [5]. Therefore, one cannot rely on the law being able to catch up with perpetrators of online crimes.

Compliance is an integral piece to the puzzle and information security is paramount at the industry level. It must be understood that IT security regulations exist for companies to not only be held accountable, but to also maintain proper data security, prevent data breaches, and minimize the financial burden when there is a data leak or loss. Cybershark, a cybersecurity outfit, states that IT regulations improve corporate security measures by setting baseline requirements [6]. It is important to note that consumers place their trust in any organization whose services they subscribe to. Maintaining compliance with these regulations is comforting to consumers. According to Cybershark, a number of US security compliance laws currently exist. While these laws may not be applicable to every industry, the most common of these regulations include the General Data Protection Regulation, Health Insurance Portability and Accountability Act, Sarbanes-Oxley Act, and Federal Information and Security Management Act of 2002 [6]. The security risks associated with outsourcing to third parties add to the overall complexity of being in compliance. While it may be difficult to maintain and know what laws or regulations apply to the services provided, they must be a priority.

3 Methodology

This project reviews compliance issues and mitigations surrounding the use of TPS applications by organizations, companies, and consumers. It is difficult to find any entity that does not rely on third parties to support its operations [7]. Compliance is considered one of the highest concerns for companies when it comes to the data they maintain or use. Compliance is defined as the set of processes, in a way that is required by a rule or law [8]. The ultimate goal of this study was to have a better understanding of the risks associated with and worth accepting when using third-party software applications and how to mitigate compliance-related issues. In addition, it is the intent to emphasize the importance of enforcing compliance— in

an effort to detect and prevent violations of procedures, which could protect major companies/organizations from litigation.

The use of literature reviews is beneficial in assessing what other entities have been observed and how to use those findings as applicable to risk management. This involves understanding the risk management process as applicable to information technology, systems, and cybersecurity.

There are a multitude of security incidents or data breaches on the rise due to the use of TPS applications. This study helps in understanding why outsourcing to a third party could pose an extreme risk and what the impacts of those risks could be. The purpose of this study is not in support of or against the use of any one particular third-party software application nor is it for or against any particular organization/entity.

4 Results

The study of compliance as it relates to security management can be defined as obedience or an agreement of the parties involved. Hackers are constantly seeking new ways to hack software systems. During the coronavirus pandemic, many Americans have resulted to doing business online, which gives hackers more opportunities to take advantage of vulnerable systems. Many Americans are considering TPS compliance being one that is equally as good as the other parties. TPS is specifically invented for businesses and other security agencies are leaning toward using it as well. The TPS has its risks—just as any security program—and can be a source of concern to security professionals who would try to improve systems daily to eliminate any confusion. The TPS has added risks especially during the pandemic period.

A few other things were revealed during this investigation regarding TPS compliance, associated with cybersecurity risks. In addition, findings confirmed that the risks associated with TPS are specifically unique to the company. This affords the opportunity to conduct trend analysis over time and determine workable steps and solutions to prevent hacking. When systems are breached, there can be panic. Panic occurs when every system is working effectively one day and then a major problem occurs the next day due to hackers. The study revealed that being prepared for the unexpected—such as the coronavirus pandemic—is important in evaluating elevated security risks. Policies, agreements, IT regulations, and requirements are implemented to ensure companies and their consumers are held to a security-focused standard. It is historically documented that organizations dedicate immense man-hours to work diligently to provide a safe and secure cloud computing/TPS environment.

5 Recommendations

The TPS compliance and its associated security risks are discussed more often since there has been an increase in services regarding third parties rendering their expertise to manage company programs [9].

When dealing with TPS, at times, companies do not handle their system with the same level of security as their own—this can expose their internal infrastructure with the vulnerabilities being exploited within that third-party application [9]. In result, this could lead to hackers accessing sensitive data within the organization or even installing ransomware—making their system inaccessible until the company pays a certain amount.

Mitigation to implement handling these risks with TPS is to conduct an analysis of the third-party software that is intended to be utilized evaluating the information security risk already identified [9]. The analysis can reveal how aligned their policies are for their customers and to ensure that the regulations are able to accurately hold the third-party companies responsible for protecting the customer's information within the software application. Additionally, the analysis also reveals the number of incidents and information security breaches that have occurred, successful and unsuccessful attempts, and the history of partnerships the third party had, which displays the outcome of those relationships. If companies pursue use of TPS with known vulnerabilities, they can implement additional controls to ensure that their company's network infrastructure is also protected [9].

In addition, companies should test the software before installing the application onto their network [10]. The software should meet security compliance requirements associated with the company's security policy and should pass all criteria listed. Anyone utilizing TPS applications should always ensure use of the latest version with the latest patch and set the requirement to automatically receive updates [10].

6 Conclusion

In conclusion, since there has been an increase in security vulnerabilities within TPS applications, a company must always be on the alert. When a company partners with a third party who offers application services, the company must have open communication of what expectations are to be met during the contract. They should sit down and go over both party's security policies to ensure they are aligned, and both agree. This prevents any disagreements in the future and allows the company to maintain their security posture to their standards. The company should always ensure that they are in compliance with their own regulations to prevent any vulnerabilities within their network but also should treat the third-party application software just as important. This ensures parties, employees, and customers are protected from cyberattacks.

References

1. Things to know about using third party cloud management ... [Online]. Available: <https://www.datacenterknowledge.com/archives/2015/10/09/things-to-know-about-using-third-party-cloud-management-tools/>. Accessed: 24-Apr-2020
2. Maintain, protect, and diminish risk with a comprehensive IT compliance strategy, Smartsheet. [Online]. Available: <https://www.smartsheet.com/understanding-it-compliance>. Accessed: 29-Apr-2020
3. J. Vijayan, Third-party breaches - and the number of records exposed – increased sharply in 2019, Dark Reading, 12-Feb-2020. [Online]. Available: <https://www.darkreading.com/attacks-breaches/third-party-breaches%2D%2D-and-the-number-of-records-exposed%2D%2D-increased-sharply-in-2019/d/d-id/1337037>. Accessed: 24-Apr-2020
4. E. Holbrook, Ensuring compliance among third parties. *Risk Manage.* **58**(4), 16–17 (2011). Accessed 27 Apr 2020
5. M.A. Rothstein, C.N. Coughlin, Ensuring compliance with quarantine by undocumented immigrants and other vulnerable groups: Public health versus politics. *Am. J. Public Health* **109**(9), 1179–1183 (2019). Accessed 27 Apr 2020
6. IT compliance: IT security regulations & compliance management, BlackStratus, 06-Sep-2019. [Online]. Available: <https://www.blackstratus.com/compliance/>. Accessed: 28-Apr-2020
7. R. Putrus, A risk-based management approach to third-party data security, risk and compliance, (2017). [Online]. Available: <https://www.isaca.org/resources/isaca-journal/issues/2017/volume-6/a-riskbased-management-approach-to-thirdparty-data-security-risk-and-compliance>. Accessed: 26-Apr-2020
8. “In compliance with,” Merriam-Webster. [Online]. Available: <https://www.merriam-webster.com/dictionary/incompliancewith>. Accessed: 26-Apr-2020
9. P. Paganimi, Third-party application security risks in modern companies, VERACODE, Oct. 15, 2015. [Online]. Available: <https://www.veracode.com/blog/2015/10/third-party-application-security-risks-modern-companies-sw>. Accessed 25 Apr 2020
10. D. Kaplan, 5 security tips for using third-party applications, Trustwave, Aug. 20, 2014. [Online]. Available: <https://www.trustwave.com/en-us/resources/blogs/trustwave-blog/5-security-tips-for-using-third-party-applications/>. Accessed 29 Apr 2020

Further Examination of YouTube’s Rabbit-Hole Algorithm



Matthew Moldawsky

1 Background

1.1 Previous Research

For many years, YouTube has been criticized by many mass media outlets for having a “right-leaning” radicalized recommendation algorithm. In other words, they claimed that the algorithm prioritized politically right channels. Last year, a report was made that sought to challenge these claims. Mark Ledwich and Anna Zaitsev [1] compiled over 800 YouTube channels that have over 10,000 subscribers, and more than 30 percent of the content is political. They set out to assess common claims from the media using a data set. They categorized the channels using different tags in order to gauge the impressions from the type of content. They had people watch content from all these channels to assign political labels through unanimous decision from the laborers. From the data that the team collected, they were able to conclude that the algorithm does not recommend content that might contribute to a radicalization of the user base. The data that they collected actually showed that the algorithm leans more toward content that falls within mainstream media. Ledwich even acknowledges the limitations of their method to their research. The research paper finishes with a conclusion that “one cannot proclaim that YouTube’s algorithm . . . is leading users towards more radical content” [1].

M. Moldawsky (✉)
Marist College, Poughkeepsie, NY, USA
e-mail: Matthew.Moldawsky1@marist.edu

1.2 *Other Important Factors and Statements*

Some important factors were not discussed in the previously mentioned paper. For instance, the recommendations for users vary depending on if they are signed in or not which the study did not cover. This caused some criticism of the paper online from various analysts, most notably from Arvind Narayanan [2]. He stated that he did not think it was really feasible to do a quantitative study of the algorithm. Another angle that should be considered is that this “radicalization” is not just about politics. For example, if a user only watches videos about certain video games, then that are mostly all of the content they will be shown. There is a clear relationship between the algorithm, users, and content creators. Radicalization, in this case, is more so the algorithm attempting to prioritize user preferences for content. The problem with this is that certain content paths can lead to extreme versions of the content a user already watches.

In late 2019, the Mozilla Foundation gathered a collection of 28 stories from various users of YouTube in which people fell down a “rabbit-hole” [3]. Lastly, it should be noted that the YouTube algorithm was designed to prioritize certain aspects of video in order to rank it. One of those aspects is engaging content as the algorithm is designed to keep users watching as much as possible. However, according to a dev who worked on the algorithm, “the more outlandish content you make, the more likely it’ll keep people watching, which in turn will make it more likely to be recommended by the algorithm” [4]. This explains why many users could be going down this “rabbit-hole” and could end up being recommended unsavory videos. This then becomes an ethical issue because this type of extreme content could be shown to children depending on the situation. Furthermore, Google, who owns YouTube, has stated previously that they wish to “recommend even more targeted content to users in the interest of increasing engagement” [5]. In other words, Google wants the YouTube platform to become more addicting by increasing the engagement of the platform. Making the platform addicting could be dangerous for children. Figure 1 shows the users across all ages of the YouTube platform.

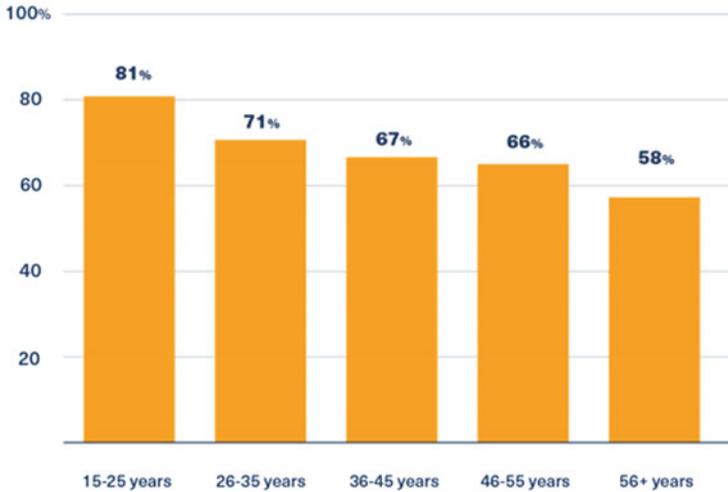
2 **Anecdotes**

2.1 *The Unintended Problems*

The following are a few anecdotes gathered by the Mozilla Foundation and then given to YouTube. Their intention was to show YouTube the problems that could arise from targeted content to increase engagement.

I started searching for “fail videos” where people fall or get a little hurt. I was then presented with a channel that showed dash cam videos from cars. At first it was minor accidents, but later it transitioned into cars blowing up and falling off

Percentage of U.S. internet users who use YouTube as of 3rd quarter 2019, by age group



Source: Statista

Fig. 1 A bar group that shows US users across a wide age range from last year [6]

bridges—videos where people clearly didn’t survive the accident. I felt a little bit sick at that point, and haven’t really sought out that type of content after that.

These terrible videos just keep being recommended to her. She is now restricting her eating and drinking. I heard her downstairs saying “work to eat! work to drink!” I don’t know how I can undo the damage that’s been done to her impressionable mind.

But my recommendations and the sidebar were full of anti-LGBT and similar hateful content. It got to the point where I stopped watching their content and still regretted it, as the recommendations followed me for ages after.

3 Conclusion

Based on the anecdotal evidence and the information given by both YouTube and Google, we can compile a list of the current problems with the YouTube algorithm. The anecdotes show that the algorithm prioritizes a user’s preferences for content. YouTube has told us that the algorithm is designed this way. The anecdotes show why YouTube’s algorithm can be described as a “rabbit-hole.” The more of a

certain type of content you watch and search for, then the more the algorithm will recommend it. Soon after searching for one type of content, you'll be recommended more. If the user clicks on that recommended video, then they will be recommended even more. The problem with this method is that some unintended problems could occur as shown by the anecdotes. The algorithm is meant to get you addicted to YouTube. The claims about radicalization are somewhat true in that users will be recommended more extreme content of what they already watch.

YouTube has already taken steps to adjust the algorithm in order to correct some of the problems many have brought up. The main problem is that the algorithm is too focused on giving a personalized experience to the user. There are some potential solutions that YouTube should investigate in order to make a healthier experience for the user. For example, instead of only recommending videos that the algorithm thinks the user might like, maybe the algorithm could include videos that other users like. Another idea is for the user to tweak their recommendations in the settings. They could prevent certain tags from showing up in recommended. The algorithm should also be tweaked to shy away from the extreme content that users may end up seeing. There will always be a case that YouTube did not account for. These measures are meant to make the algorithm healthier in general.

Acknowledgments I would like to express my thanks to Dr. Pablo Rivas of Marist College who supervised the research and review of this paper. Whenever I needed guidance during the research and writing process, you were there to help. Once again, thank you for your support.

References

1. M. Ledwich, A. Zaitsev, Algorithmic extremism: Examining Youtube's rabbit hole of radicalization, <https://arxiv.org/pdf/1912.11211.pdf>. 19 February, 2020
2. H. Kozłowska, Does YouTube favor radicalization? From outside YouTube, it's hard to know. <https://qz.com/1777381/its-hard-to-know-if-youtubes-algorithm-promotes-radicalization/>. 10 April, 2020
3. "YouTube Regrets," <https://foundation.mozilla.org/en/campaigns/youtube-regrets/>. 28 April, 2020
4. M. Maack, 'YouTube recommendations are toxic,' says dev who worked on the algorithm, <https://thenextweb.com/google/2019/06/14/youtube-recommendations-toxic-algorithm-google-ai/>. 20 February, 2020
5. K. Hao, YouTube is experimenting with ways to make its algorithm even more addictive, <https://www.technologyreview.com/2019/09/27/132829/youtube-algorithm-gets-more-addictive/>. 18 March, 2020
6. P. Cooper, 23 YouTube statistics that matter to Marketers in 2020, <https://blog.hootsuite.com/youtube-stats-marketers/>. 18 March, 2020

Part VII
Educational Frameworks and Strategies,
and e-Learning

Characterizing Learner's Comments and Rating Behavior in Online Course Platforms at Scale



Mirko Marras and Gianni Fenu

1 Introduction

The development of a successful career lies with the individual's ability to continuously acquire knowledge, gain competencies, and get qualifications. Online learning platforms at large scale can support individuals toward achieving these goals, as an ecosystem wherein people, resources, technology, and processes interact to train lifelong learners, without time or place constraints [11]. Notable examples include *Coursera* and *Udemy*, which host learners and instructors within thousands of courses. Scaling up education online is posing key challenges related to overwhelming content alternatives and huge amounts of data to analyze [12].

Specifically, online course platforms at large scale act as social spaces where learners consume content and express opinions about courses [13]. The expressed opinions (e.g., comments or ratings) often convey a positive or negative polarity according to the writer's satisfaction for that course. Such a collective intelligence might be useful for peers planning to attend a course, instructors interested in improving their teaching, and policy makers who want to understand learners' satisfaction [2, 9, 14]. Furthermore, learner's opinions reveal insights that serve as an input for other purposes, such as the improvement of educational services for recommendations, the design of changes in the platform, and the refinement of teaching practices based on sentiment and affective computing [1, 5, 10].

When a characterization of learner's opinion patterns was conducted in prior work, the focus was on a specific course or program [8]. Other studies, instead, extracted satisfaction indicators from classroom sensing data [15] or surveys provided by a single university [7] and analyzed the influencing factors on forum

M. Marras (✉) · G. Fenu

Department of Mathematics and Computer Science, University of Cagliari, Cagliari, Italy
e-mail: mirko.marras@unica.it; fenu@unica.it

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_56

781

posts [4] or personality and satisfaction in online academics [3]. With our study, this is the first time that learner's opinions are characterized in online learning platforms at scale, with a *worldwide user base*, considering several perspectives.

In this study, we characterize learners' tendencies, while they express opinions in the form of textual comments or ratings in this emerging scenario. We explore dynamics on 4,618,113 opinions released by 3,234,004 learners on 27,711 online courses, offered by 12,218 instructors over the last 10 years. Unlike academic-oriented coursework, large-scale platforms enable experts to offer courses in free or pair mode. No third-party control on reliability and truthfulness of the course is usually performed. Therefore, it becomes essential to understand the enormous amount of opinions and assess how learners perceive the content being provided online. Our study will focus on the following three perspectives:

- **Course.** This perspective allows us to study what the learners look for and the behavior associated with course selection and evaluation (e.g., courses likely to be rated and temporal aspects associated with course opinions).
- **Learner.** This perspective gives us insights on the individual's tendencies in opinion delivering and on the habits of learners while rating courses. Different learners may have different evaluation habits and scoring dynamics.
- **Instructor.** This perspective focuses on characterizing how opinions vary, based on the instructor who receives them, a key stakeholder in this context, mining relations between instructor's characteristics and opinion patterns.

To the best of our knowledge, there exists no other characterization of opinions that includes such an extent of educational ratings and comments from large-scale online courses. Our contributions can be thus summarized as follows:

- We conduct, for the first time in the literature, a study of the learner's opinions and rating behavior in a massive open online course platform.
- This is the first study giving a multi-stakeholder perspective (i.e., learners, instructors) to opinion analysis in online course platforms at scale.
- We provide findings useful for the research community working on online learning, with practical implications for the design of educational services.

2 Dataset Description

The analyzed dataset has been originally provided in [6]. It comes from an online learning marketplace that allows professionals and companies to provide learners with online courses. Courses are categorized according to a two-level taxonomy, with a primary and a secondary category assigned to each course.

The dataset includes 27,711 courses published between *Jan 2010* and *Oct 2017*. Each one is described by an *id*, a *title*, a *short*, and a *long description*. *Requirements* and *objectives* report the knowledge required at the beginning and

expected competencies at the end of the course, respectively. The *language* (out of 15), the *instructional level* (out of 4), and the *releasing date* are included.

The 12,218 *instructors* and the 3,234,004 *learners* are uniquely identified by an anonymized *id*. Each learner and instructor is described by his/her *time zone*, *local language*, and *signup date*. Each instructor gives one or more courses, and the same course has one or more instructors. Learners can release an opinion for a course they have attended. The original dataset in [6] has been extended to include opinions released by learners between *Jan 2010* and *Jan 2020* on the same courses. Each of the 4,618,113 opinions is specified by a *textual comment*, a *rating score* between 0.5 and 5.0 with a step of 0.5, and an ISO-formatted *timestamp* associated with the time the opinion has been released.

3 Characterizing Opinions from a Course Perspective

In this section, we characterize learners’ opinions from the perspective of courses. To this end, we answer three main research questions:

- RQ1. To what extent are courses receiving opinions from learners? In what form?
- RQ2. Do the amount and type of opinion vary, based on the course category?
- RQ3. How do course opinions vary based on the average rating score of a course?

Opinion Modality Analysis (RQ1) In order to characterize the tendency of learners to give opinions on courses, we grouped the opinions based on the year and month they were released, according to the timestamp field, and counted how many opinions appear on average for a course in each group. Figure 1a presents the results in a time plot. The results show us that learners are more and more inclined to share opinions on a course. This might reveal that learners consider important to provide opinions on their past experiences, so that future learners can make more informed enrollment decisions. From instructor and policy maker perspectives, an increasing number of opinions per course would support a better assessment of the education

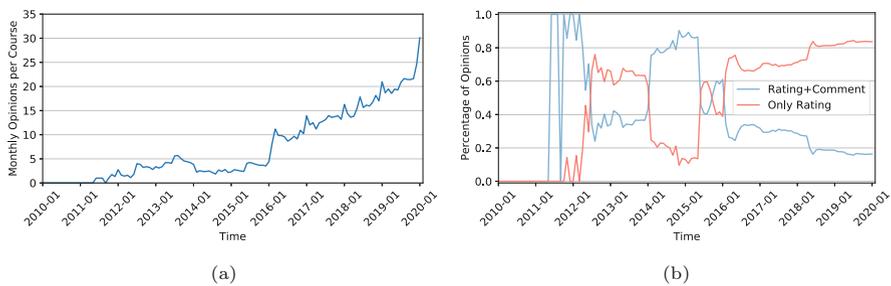


Fig. 1 Opinions Along Time. Temporal patterns behind opinions on online course. (a) Avg. monthly opinions per course along time. (b) Opinion granularity per month along time

quality. Indeed, with more and more opinions, an automated support provided by sentiment and affective analyzers will become essential. As these systems capitalize on opinions to learn patterns, it is interesting to see how learners prefer to convey their opinions. Figure 1b shows, for each year and month, the percentage of opinions including both a rating and a comment or only a rating, given the total number of opinions given in that period. Learners used to share richer opinions in past years, while the last 4 years highlighted an evident tendency toward ratings only. It follows that, as learners give only a rating, assessing the reasons behind that score becomes harder.

Opinion per Course Category Analysis (RQ2) In order to characterize how the amount and type of opinion impact on courses based on their category, Fig. 2a presents two bars per category. The yellow bar characterizes the percentage of courses from that category with respect to the total number of courses, the light blue bar indicates the percentage of opinions for courses of that category with respect to the entire set of opinions. Results show us that categories are differently distributed in the catalog and the opinions. The most represented categories in the catalog experienced an increment in representation in the opinion set. Conversely, courses belonging to the less represented categories ended up receiving few or no opinions. It follows that the latter courses may be offered to few learners in the future, even if they are of interest, if educational services do not pay attention to such an imbalance. It becomes important also to ask us if and how the type of opinion changes based on the course category. Hence, Fig. 2b shows, for each category, the percentage of opinions including both a rating and a textual comment or only a rating. Learners who attend courses in computer science-related fields (i.e., *IT & Software*) provide less opinions with a comment, i.e., around 19%. Conversely, courses on other topics (e.g., *Fitness and Photography*) proportionally received more textual comments.

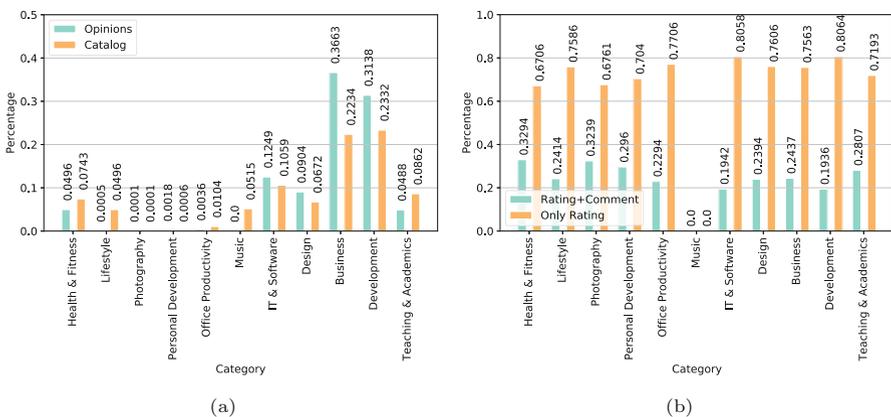


Fig. 2 Opinions per category. Proportion and granularity of opinions based on the course category. (a) Course category proportion. (b) Opinion granularity per course category

Opinions Based on the Average Rating Score of a Course (RQ3) In order to understand the characteristics of opinions with respect to the rating behavior of learners, we first grouped courses based on the average rating score they received and counted how many opinions learners released for each group. Figure 3a shows a direct relation between the course quality recognized by learners (i.e., average rating) and the number of opinions released for those courses, during the first months after the course publication. It follows that courses receiving high rating scores by the time they were published are likely to be attended by more learners and thus receive more opinions. We then investigated whether there is any pattern in the type of opinion according to the average rating of a course. Figure 3b shows the percentage of opinions with comments and opinions with only ratings for courses grouped based on the average rating score. Learners tend to give comments when courses are associated with low rating values. It seems that learners use comments to highlight course drawbacks and not for describing their experience as a whole. In Fig. 3c, we also reported the average number of words included in a comment, for courses grouped by their average rating. Results show us that learners often provide short comments and that courses with high average rating scores tend to receive shorter comments.

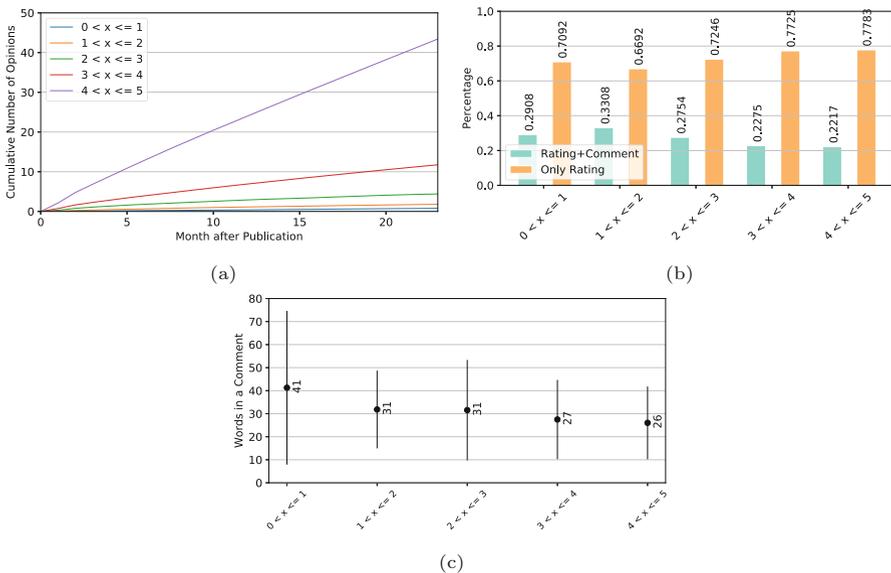


Fig. 3 Opinion-rating relation. Statistics on opinions for courses grouped by average rating score. (a) Opinions along time for courses. (b) Opinion granularity for courses. (c) Words in a comment for courses, grouped by avg. rating. Bullets represent the average number of words; black lines indicate their std. deviation

4 Characterizing Opinions from a Learner Perspective

The opinion behavior of the learners at individual level allows us to understand the aspects that are important for them, when they provide opinions. With this analysis, we aim at answering the following research questions:

- RQ4. How do learners give ratings to courses over rating scale? How over time?
- RQ5. Is there any pattern in opinions based on how many courses learners rate?

Learners' Rating Score Analysis (RQ4) In order to answer to RQ4 and understand how learners use the rating scale, we count the number of times each rating value admitted by the rating scale is being assigned by learners. As shown by Fig. 4a, while learners are provided with ten possible alternatives of rating a course, they often end up assigning the maximum rating score (i.e., 5-star). Given this imbalance among values in the rating scale, Fig. 4b shows the percentage of opinions with a rating equal to or lower than the maximum value of 5, for each year and month. Learners used to assign high rating scores until 2016, while a more equal proportion was achieved in last years, showing a clear shift in evaluation patterns. Systems, such as recommenders and sentiment analyzers, might consider recent data, which provides more diversified rating scores, if such scores are needed to drive model optimization (e.g., rating predictors).

Learners' Opinion Contribution (RQ5) In Fig. 5a, we characterize how learners individually contribute to the opinion set, grouping them based on the number of opinions they released. The 86% of the learners provided only one opinion. This result could reveal that learners tend to rate and/or attend courses occasionally and poses a challenge for those systems that are optimized on top of opinion data (e.g., recommenders). To have a detailed picture, Fig. 5b shows how many opinions each learner group provided for each admitted rating score. Learners who released only one opinion used to assign lower rating scores, possibly implying that learners who experienced low-quality courses end up dropping out of the

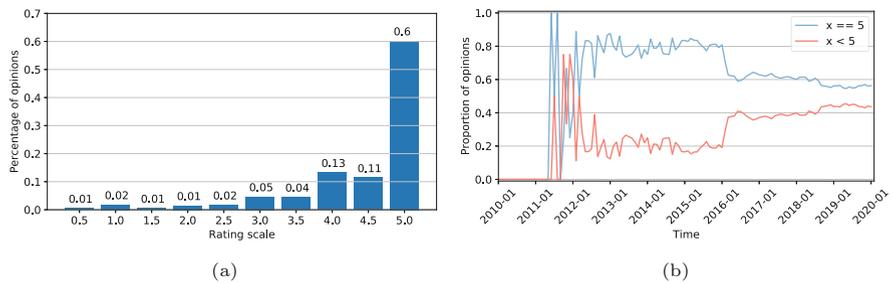


Fig. 4 Opinions along the Rating Scale. Learner's tendency to use the alternative values included in the rating scale. (a) Proportion of ratings for each rating value. (b) Proportion of 5-star ratings and other ratings

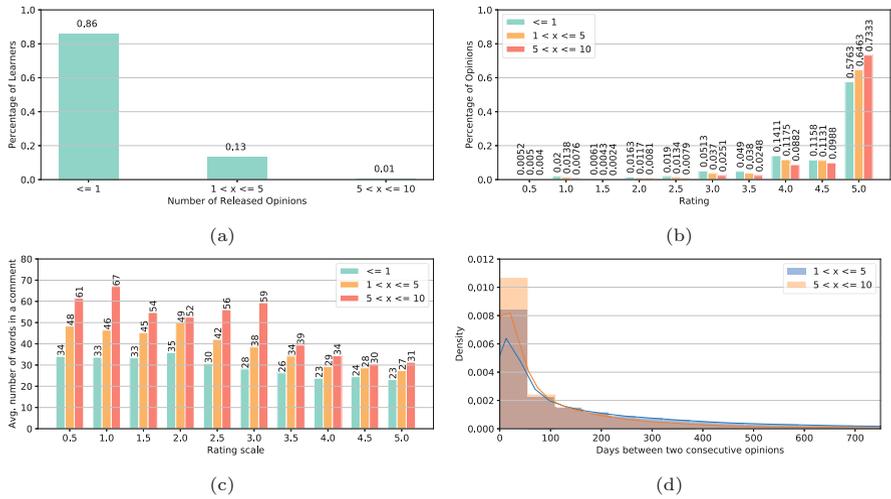


Fig. 5 Opinions Across Learner Groups. Representative statistics on learner’s opinion delivering, based on the number of courses the learner rated. (a) Learners grouped by a number of rated courses. (b) Opinions along the rating scale. (c) Comment length along the rating scale. (d) Days between opinions, per learner

platform. Thus, individual support might be deserved for them. Figure 5c reports the average number of words included in a comment provided by learners belonging to the considered groups. Learners who provided only one opinion ended up writing shorter comments. Furthermore, different levels of participation also emerge from Fig. 5d, since learners who attended more courses used to wait less time before starting another course.

5 Characterizing Opinions from an Instructor Perspective

It is also important to understand how instructors are impacted by the opinions released by learners. Here, we analyze the opinion behavior, considering:

- RQ7. How do opinions differ based on the number of courses an instructor gave?
- RQ8. Do opinions convey different patterns based on the instructor language?

Opinion Representation per Instructor (RQ7) We selected the oldest course of each instructor and grouped those courses based on the month they were published. Figure 6a depicts the number of new instructors who joined the platform, monthly. From 2016, the number of newcomers per month regularly ranged between 200 and 300 new instructors. We then analyzed how many courses each instructor published on the platform in Fig. 6b. While the clear majority of instructors used to provide only one course, the instructor’s contribution to the platform based on the number of

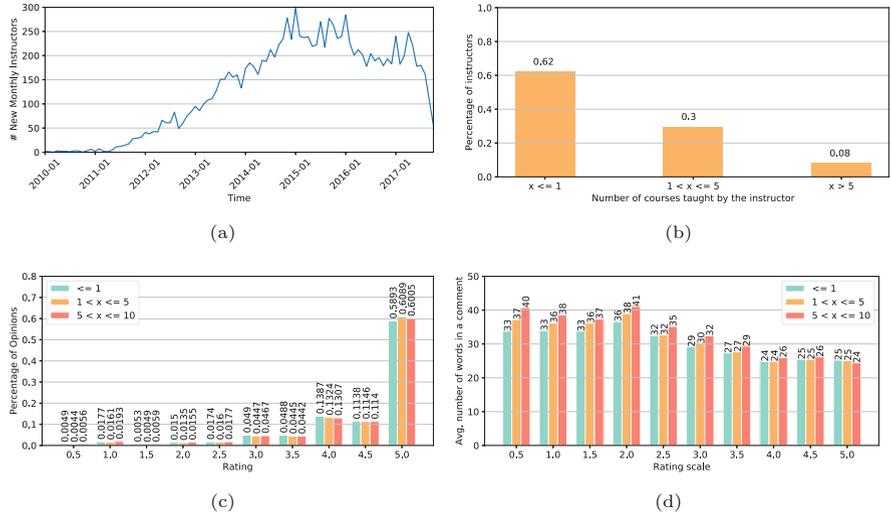


Fig. 6 Opinions Along Instructor Groups. Representative statistics on learner’s opinion delivering, based on the number of courses the instructor gave. (a) New instructors, monthly, along time. (b) Instructors grouped by the number of given courses. (c) Opinion percentage along rating scale. (d) Comment length along rating scale

given courses is more balanced, with respect to the learner’s contribution based on the number of rated courses (Fig. 5d). Given such a grouping of instructors, Figs. 6c and d, respectively, show the proportion of opinions and the average comment length along rating scores, for each instructor group. There is no statistically significant difference (paired t-test, $p = 0.05$) on the proportion of opinions among groups. On the other hand, shorter comments are received by instructors, when courses receive low rating.

Opinion Analysis Based on Instructor Language (RQ8) Online learning at scale allows instructors, from diverse countries, speaking different languages, to reach a wider audience (Fig. 7a). It thus becomes relevant to investigate if and how opinions vary based on the language used to give the course, which might be a source of imbalance and, thus, of possible biases for data-driven educational services. To this end, Fig. 7b shows the proportion of courses and opinions based on the main language of the course. Results show us that original proportions in the catalog are kept to a good extent also on the opinion set, with small variations for *English*, *German*, and *Portuguese*. Conversely, Fig. 7c and d sheds light on differences across languages with respect to the type of received opinion and the average number of words in the comments. In Fig. 7c, while similar proportions of opinions with a comment are observed for certain languages (e.g., *English* and *Spanish*), there exist courses receiving a smaller (e.g., *Portuguese*) or larger (e.g., *Japanese*) proportion of comments. Indeed, courses receive shorter or longer comments based on the

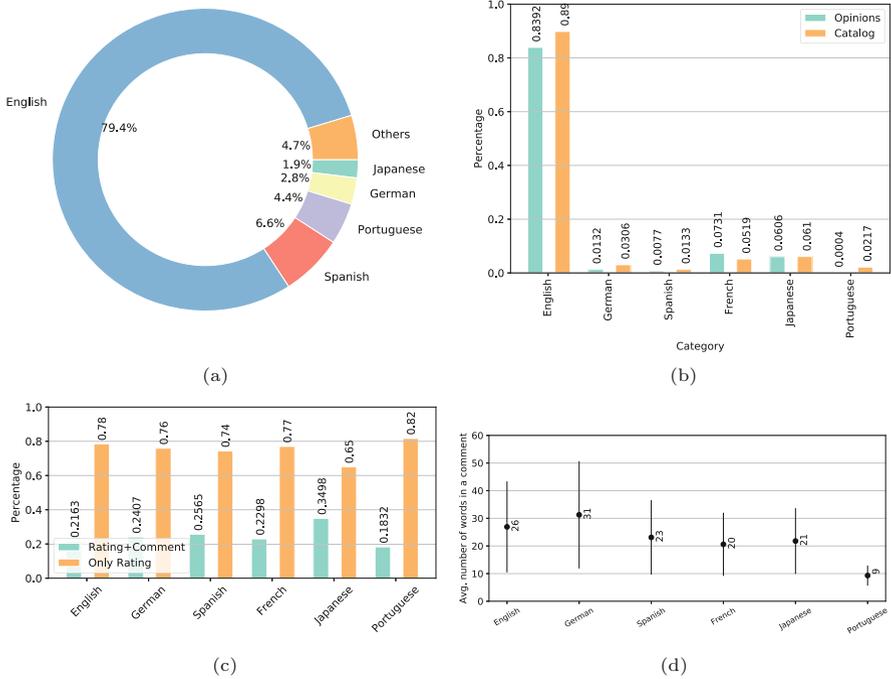


Fig. 7 Opinions Along Instructor Language. Representative statistics on learner’s opinion delivering, based on the course language. (a) Instructor distribution across languages. (b) Opinion/course proportion per language. (c) Opinion granularity per language. (d) Avg. number of comment words per language

language of the course, with *Portuguese* experiencing the lowest average number of words (Fig. 7d). With the last examples provided in this analysis, we were interested in highlighting the importance of considering imbalanced situations in this domain. With imbalances, data-driven systems might perform differently, based on sensitive characteristics of individuals (e.g., instructors), potentially leading to biased or even unfair situations.

6 Discussion

In this section, we connect the insights coming from our analysis, discuss the results, and provide take-home messages.

Large-scale course platforms are attracting lots of learners, and their interaction is generating a vast amount of data. Hence, an exploratory analysis, as proposed by this study, is essential to understand key patterns in data collected from online learning activities at scale, not likely to be visible externally. With our findings,

researchers can make more informed decisions when selecting an approach for tackling their research question and identify new research questions. For instance, implicit and explicit feedback associated with learners is very sparse, given that attending a course is more time-consuming than listening to a song or watching a movie, as examples. Furthermore, researchers in recommender systems may opt to select algorithms based on implicit feedback for this domain, due to the imbalanced rating distribution (Sect. 4). This study sheds light also on patterns and biases related to the category and the language of a course, which need further investigation. Similarly, researchers interested in sentiment and affective analysis in this domain should consider that comments given by learners are shorter with respect to the ones given in other domains, such as tweets or product reviews (Sect. 3). Indeed, our insights on comment length, ratings, and language shed light on the need of a fine-grained data preparation step for training models as well as on evaluation protocols that consider the variability of opinions over such dimensions (Sect. 5). Such observations may, in turn, influence other educational services as well.

With this study, we showed peculiar characteristics of data collected in an online learning scenario at scale. It may be inevitable that, as systems empowered with user-generated data move further into education, it will become more and more necessary to consider aspects as the ones we presented.

7 Conclusion and Future Work

In this chapter, we characterized opinion dynamics in the context of online learning at scale. Specifically, we investigated temporal aspects related to how and to what extent learners share their opinions and highlighted differences based on comment and rating patterns. Finally, we connected different perspectives and envisioned practical implications on data-driven educational systems.

Our results showed that learners are more and more likely to share opinions on the courses they attend, so each course is expected to receive more feedback that helps to improve learning and teaching practices. However, learners tend to provide only ratings, not textual comments. The amount and length of comments depend on the rating score, with more and longer comments for courses that receive low rating scores. Our analysis revealed that, differently from other domains, the distribution of ratings is imbalanced toward high rating values, and opinions are highly sparsed (i.e., each learner gives only few opinions, and a lot of courses receive a small number of opinions). More active learners provide longer comments, highlighting the key role of educational platforms as a social space for sharing course opinions as well. Finally, we showed that instructors are active stakeholders in this domain, with half of the them providing more than one course. The amount and type of opinion they receive depend on the average rating scores of their courses and the language they used to provide content.

Future work will extend this study with insights related to learners and instructors, based on their personal attributes, enabling us to profile them along gender

and age, as an example. An important factor to characterize is the learners' attitude toward opinions, such as why they decide to evaluate a given course. To move a step forward in this direction, future work will enrich the analysis to identify factors across learners' socio-demographics, with key implications on multiple design aspects (e.g., the fairness of automated educational models).

Acknowledgments This research is partially funded by Sardinian Regional Government, POR FESR 2014-2020 - Axis 1, Action 1.1.3 - Project SPRINT (D.D. n. 2017 REA del 26/11/2018, CUP F21G18000240009).

References

1. L. Boratto, G. Fenu, M. Marras, The effect of algorithmic bias on recommender systems for massive open online courses, in *Proceedings of the European Conference on Information Retrieval, ECIR* (Springer, Berlin, 2019), pp. 457–472
2. G.S. Chauhan, P. Agrawal, Y.K. Meena, Aspect-based sentiment analysis of students' feedback to improve teaching–learning process, in *Information and Communication Technology for Intelligent Systems* (Springer, Berlin, 2019), pp. 259–266
3. A. Cohen, O. Baruth, Personality, learning, and satisfaction in fully online academic courses. *Comput. Human Behav.* **72**, 1–12 (2017)
4. A. Cohen, U. Shimony, R. Nachmias, T. Soffer, Active learners' characterization in MOOC forums and generated knowledge. *J. Educ. Technol.* **50**(1), 177–198 (2019)
5. D. Dessì, M. Dragoni, G. Fenu, M. Marras, D.R. Recupero, Evaluating neural word embeddings created from online course reviews for sentiment analysis, in *Proceedings of the ACM/SIGAPP Symposium on Applied Computing, SAC* (2019), pp. 2124–2127
6. D. Dessì, G. Fenu, M. Marras, D.R. Recupero, Coco: Semantic-enriched collection of online courses at scale with experimental use cases, in *Proceedings of the World Conference on Information Systems and Technologies, WorldCist* (Springer, Berlin, 2018), pp. 1386–1396
7. V. Dyomin, G. Mozhaeva, O. Babanskaya, U. Zakharova, MOOC quality evaluation system: Tomsk state university experience, in *Proceedings of the European Conference on Massive Open Online Courses* (Springer, Berlin, 2017), pp. 197–202
8. G. Elia, G. Solazzo, G. Lorenzo, G. Passiante, Assessing learners satisfaction in collaborative online courses through big data approach. *Comput. Human Behavior* **92**, 589–599 (2019)
9. P. Gómez-Rey, E. Barbera, F. Fernández-Navarro, Measuring teachers and learners perceptions of the quality of online learning experience. *Distance Educ.* **37**(2), 146–163 (2016)
10. P.V. Kulkarni, S. Rai, R. Kale, Recommender system in elearning: A survey, in *International Conference on Computational Science and Applications* (Springer, Berlin, 2020), pp. 119–126
11. R. Panigrahi, P.R. Srivastava, D. Sharma, Online learning: Adoption, continuance, and learning outcome - a review. *Inter. J. Info. Manag.* **43**, 1–14 (2018)
12. G. Pike, H. Gore, The challenges of massive open online courses (MOOCs), in *Creativity and Critique in Online Learning* (Springer, Berlin, 2018), pp. 149–168
13. V. Psyché, B.K. Daniel, J. Bourdeau, Learning spaces in context-aware educational networking technologies in the digital age, in *Educational Networking* (Springer, Berlin, 2020), pp. 299–323
14. S. Rani, P. Kumar, A sentiment analysis system to improve teaching and learning. *Computer* **50**(5), 36–43 (2017)
15. Y. Zhonggen, Z. Ying, Y. Zhichun, C. Wentao, Student satisfaction, learning outcomes, and cognitive loads with a mobile learning platform. *Comput. Assisted Language Learn.* **32**(4), 323–341 (2019)

Supporting Qualification Based Didactical Structural Templates for Multiple Learning Platforms



Michael Winterhagen , Minh Duc Hoang, Benjamin Wallenborn ,
Dominic Heutelbeck , and Matthias L. Hemmje 

1 Introduction

Higher Education Institutes (HEI) often use a *Learning Management System* (LMS) to provide learning content for the learners. Related to the learning content of HEIs there are three actors [10]: the HEI, the consumers of learning content, and the producer of learning content.

Producing learning content for HEI consumers is quite complex in LMSs for producers of learning content. The *Knowledge-Management Ecosystem Portal* (KM-EP) [5] contains an Educational Portal subcomponent which contains different educational subcomponents.

A *Course Authoring Tool* (CAT) has been introduced in [10] within the KM-EP to reduce the complexity of a Moodle [8] course production and make it simple to produce courses in the way that the producers only have to fill in the necessary information and are able to concentrate in producing the learning content instead of configuring the learning content within the LMS.

M. Winterhagen (✉) · M. L. Hemmje

Chair of Multimedia and Internet Applications (MMIA), University of Hagen, Hagen, Germany
e-mail: michael.winterhagen@fernuni-hagen.de; matthias.hemmje@fernuni-hagen.de

M. D. Hoang · B. Wallenborn

Center for Media and IT (ZMI), University of Hagen, Hagen, Germany

e-mail: minh-duc.hoang@fernuni-hagen.de; benjamin.wallenborn@fernuni-hagen.de

D. Heutelbeck

Forschungsinstitut für Telekommunikation und Kooperation e.V. (FTK), Dortmund, Germany

e-mail: dheutelbeck@ftk.de

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_57

793

1.1 Motivation, Problem Statement, and Approach

Within the CAT it is also possible to save so-called course templates. These templates can be empty courses structures or completely produced courses (with the complete learning content for this course) and must be cloned and filled with the desired learning content. So, the producers have to create a course template only once, and it can be reused in multiple implementations of this course structure.

Although having a possibility to handle course templates and to create courses easier with the CAT, the course templates do not capture the pedagogical structure of a course. The course templates just reflect the logical structure of a course.

A second problem with the templates within the CAT of the KM-EP is that they are implemented for the LMS Moodle and cannot be exchanged with another system that does not support Moodle course formats. Therefore, it is necessary to have a more abstract definition of pedagogical templates, which can be exchanged between different systems to support interoperability and system integration.

Therefore, to support course authoring it would be nice to have a function to define the pedagogical structure of a course as a pedagogical template and reuse this pedagogical structure as a template for courses with the same pedagogical structure.

Therefore, the so-called *Didactical Structural Templates* (DST) have been introduced in [11] as *Structural Templates* (ST) which represent the structure of a course—and have been extended to represent the pedagogical structure of a course in [12].

The DSTs base on the *IMS Learning Design* (IMS-LD) [1] and are combined with the *Qualifications-Based Learning Model* (QBLM) [9, 10].

The IMS Global Learning Consortium introduced the *IMS Learning Design* (IMS-LD) [1]. The IMS-LD specifies different elements, which stay in a hierarchical relationship. Every IMS-LD consists of a method, which contains one or more pedagogical structures. The hierarchical part below the pedagogical structure, a so-called play [1] from IMS-LD, is called *Learning Path* (LP) [1].

The concept of an LP is used as a base for the pedagogical structure. With this concept we can differentiate between the learning content and the pedagogical structure.

Having the DSTs as an abstract definition of a LP and as pedagogical structure, these DSTs cannot only be used as pedagogical structure for creating courses. In fact, the DSTs can also be used as a pedagogical structure for a hybrid environment existing of a “classical” course with integrated applied gaming content just like a pedagogical structure for applied game, which can be a web-based computer game or a VR/AR based game, therefore one DST can have different implementations.

The advantage of this approach is that learners will be able to switch between different implementations of one DST whenever they want to, and they have got the same learning progress as if they had used only one specific implementation of this DST. This means if learners like gaming, they can use the applied gaming implementation to work on the learning content. If it is easier for the learners to answer the self-tests or the final test—to stay in the exemplary stated pedagogical

structure of a course—as e.g. multiple-choice quizzes, they can switch to a course within a traditional LMS to answer the questions.

The concept of *Qualifications-Based Learning* (QBL) has been introduced in [9] and [10]. The domain model of QBL is called a *Qualifications-Based Learning Model* (QBLM). It serves as a flexible base with which it is possible to define standardized, machine-readable qualifications/competences and bundle them up within a *Competence Framework* (CF). For the different meanings see [9]; for the sake of simplicity, the term competence is used below as a synonym for competence and qualification. A Competence Profile (CP) is a bundle of competences and in our case is also a subset of a CF.

As described above, the KM-EP contains different management systems. One of these educational subsystems is the so-called Content and Knowledge Management System, which contains among other managers a *Competence Manager* (CM). With this CM, it is possible to manage the CFs.

In order to make it possible to define didactic goals in the form of competences they already are in the DST and to have the possibility to check, whether a learner can take part in a specific learning element or not, we have to extend the IMS-LD specification with the QBLM approach.

Therefore, the CPs in our approach are used to define prerequisites and Learning Goals (LG) of courses, but only on the level of courses. With this, it would be possible to check, if the learner can take part in this specific course, but not, if he can take part in specific course/learning elements. But this is needed to enable the above-described possibility to switch between the different implementations of a DST. Therefore, the QBL approach has to be extended to make this goal possible. To achieve this, we use the concept of learning elements, which are introduced in [9].

To summarize the remainder of this paper addressing our research questions, which we will require to work on shown below, is: *How can the DSTs be accessed from any kind of tool or production environment?*

2 State of the Art in Science and Technology

In order to make the DSTs accessible any kind of tool or production environment, we will describe two options:

1. export the DST as file, and
2. provide an API for and kind of tool.

In this section we will describe, what the state of the art is like. After this is done, we will show, which open challenges exist.

2.1 *Export of DSTs*

We already described in [12] that the DSTs are based on the IMS-LD. As described in Sect. 1.1, the QBL approach is a flexible base with which it is possible to define standardized, machine-readable qualifications/competences and bundle them up within a CF. Therefore, it is a good choice to use this approach as a base of prerequisites and goal competences of the learning elements of the DST.

As far as we know, there is currently no editor for the IMS-LD under development. Even if there is still an editor under development, it will not support the QBL approach. Therefore, we have to develop our own editor and a corresponding management system which we will call *Didactical Structural Template Manager* (DSTM). Therefore, the option to export the DST as a file will be placed within the DSTM.

Because of the fact, that the DSTs are based on the IMS-LD, we will extend the IMS-LD specification by the QBLM approach.

The IMS-LD offers three specifications in XML format, which are built upon each other. They are namely *Level A*, *Level B*, and *Level C*. For our purpose, we only need the *Level A* specification. Because of the fact that all three specifications are built upon each other, our extension will also be contained within the other specifications *Level B* and *Level C*.

2.2 *Providing an API*

The Learning Design API is defined as a *Representational State Transfer* (REST) [4] or respectively a RESTful interface, because it is easy to implement and easy to access. The return value of each endpoint is made in *JavaScript Object Notation* (JSON) [3].

We described in the sections before that the open task is how the IMS-LD specification has to be extended. This challenges will be followed in the remaining of the paper.

3 **Conceptual Work**

As described in the section before, we have to extend the IMS-LD specification. In this section, we will show what has to be done to extend this specification.

3.1 Application Use Cases

In the application context, we have three actors, which relate to the DSTM ([7]):

- The *Competence Manager* is usually a high-level educator who should have extensive knowledge of QBL. The most important mission of a competence manager is the design of CFs and CPs.
- The *Learning Designer* should take care of all Structural Templates. More specifically, the users of this role can create a new DST and design it by inserting the required structural elements of IMS-LD into this DST. In addition to the design process, Learning Designers can assign any existing competency profile as a prerequisite or goal for items in a DST.
- The *Teacher* is the traditional role of every learning system. In the current state of development of the KM-EP, the teacher can use CAT to manage existing courses and add learning activities and collaboration services to each course.

The following figure shows the different use cases for the three described actors (Fig. 1):

3.2 Extension of the IMS-LD Specification

What is missing in the IMS-LD specification is the possibility to add CPs and CFs. Therefore, we have to extend the existent IMS-LD specification. A rough overview of the existing IMS-LD specification Level A (Fig. 2):

The QBLM structure has to be placed within the IMS-LD specification. This is done by extending the XML element *Component* the following way (elements within the marked area) (Fig. 3):

Using this possibility for an external tool or as a structure of an applied game, it is necessary to export the structure of a DST as a file and to import this pedagogical structure to make the pedagogical structure reusable.

4 Prototypical Implementation

After defining the extension of the IMS-LD specification to make it possible to provide the DST as a file, and defining how we want to provide the DSTs via API, we will show in this section how we define our API.

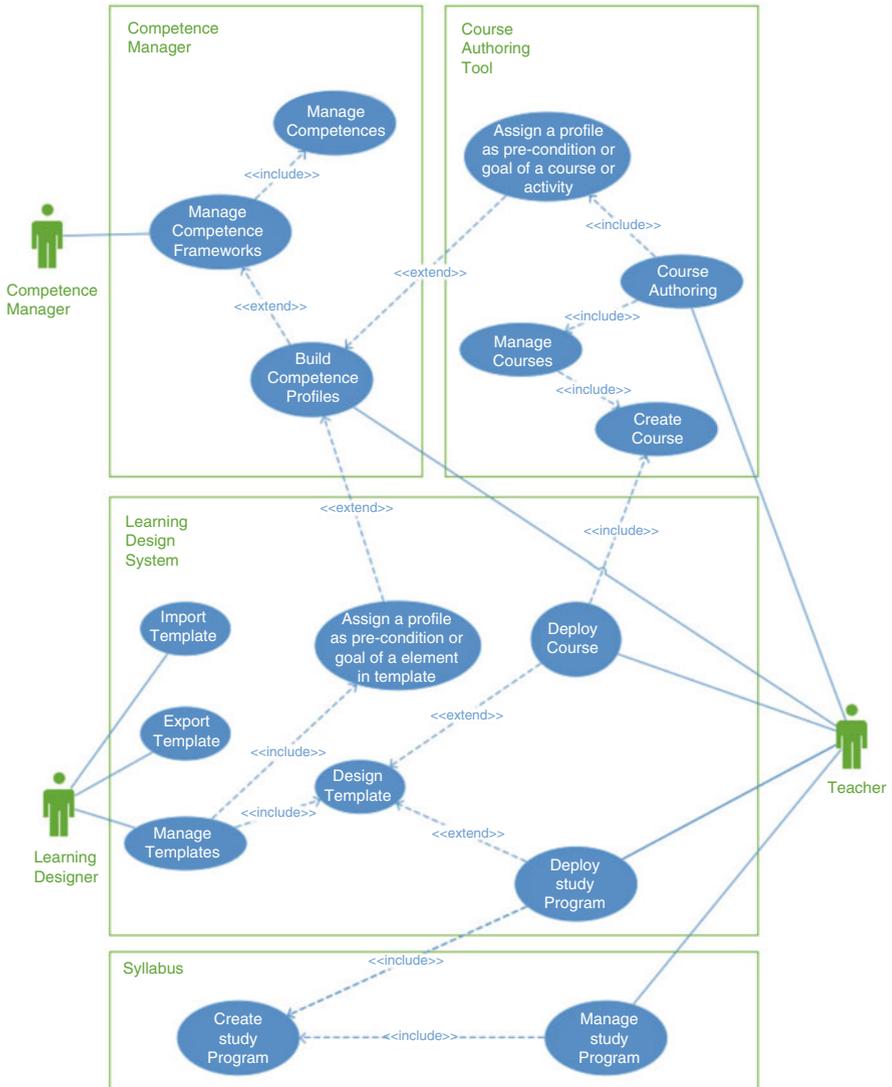


Fig. 1 Use case diagram for the usage of the DSTM [7]

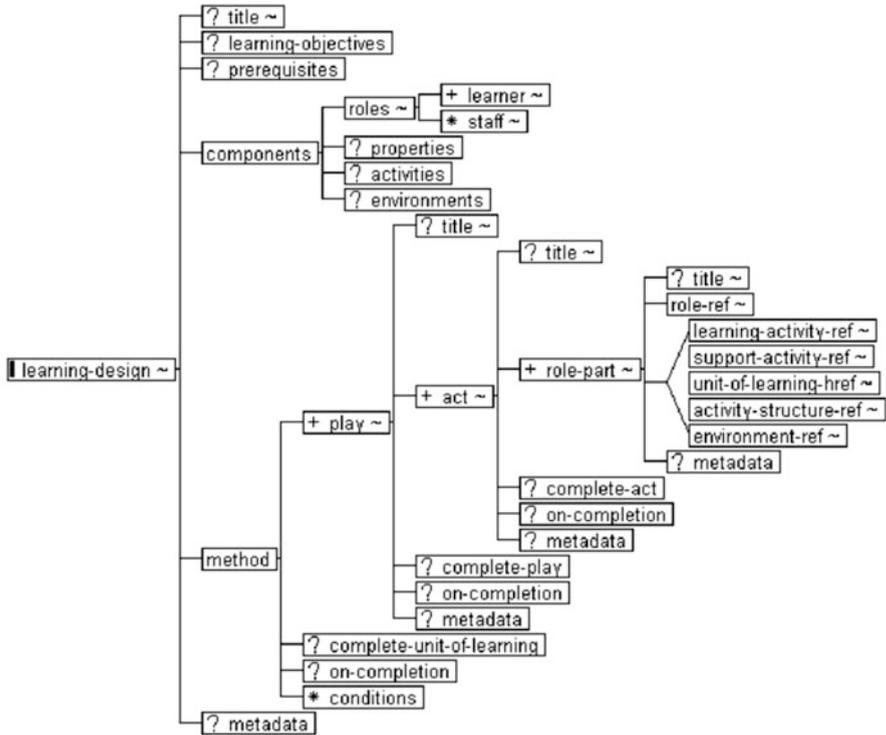


Fig. 2 The XML schema tree of IMS-LD [7]

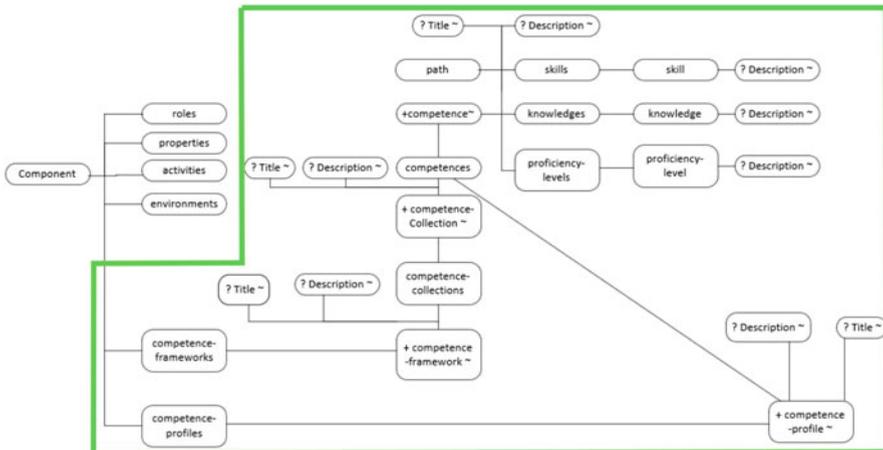


Fig. 3 The new XML schema tree of component [7]

4.1 Implementation of the Didactical Structural Template Manager

As we described in Sect. 1, the KM-EP Educational Portal subcomponent which contains different educational subcomponents. We also described in Sect. 2.1 that we want to implement the *Didactical Structural Template Manager* (DSTM) and include it into the KM-EP.

We have the following functional requirements to be realized by the DSTM (subset of the functional requirements described in [7]):

- *Design a DST*: Designer can create a DST and add Play to it. After that, Act can be designed with Activities and Activity Structures. Each element in this hierarchical structure of Learning Design can be edited and deleted.
- *Assign an CP*: A CP can be used to describe goal and condition of Learning Design or its elements.
- *Wrap a DST and export*: Each DST can be exported as a file which will be used not only in other deployments of KM-EP to recover this DST, but also exploited in other Authoring Tools of Learning Design.
- *Import DST from an exported file*: The exported files should be imported to create the same DST with classification and competency information.
- *Install a DST*: A study program could be created from a DST.

The integration of the DSTM (still called Structural Template Manager in the figure) within the KM-EP is shown in Fig. 4. The other subcomponents which are shown in this figure will not further be explained in this paper.

The functionality of the DSTM will be shown in Chap. 5.

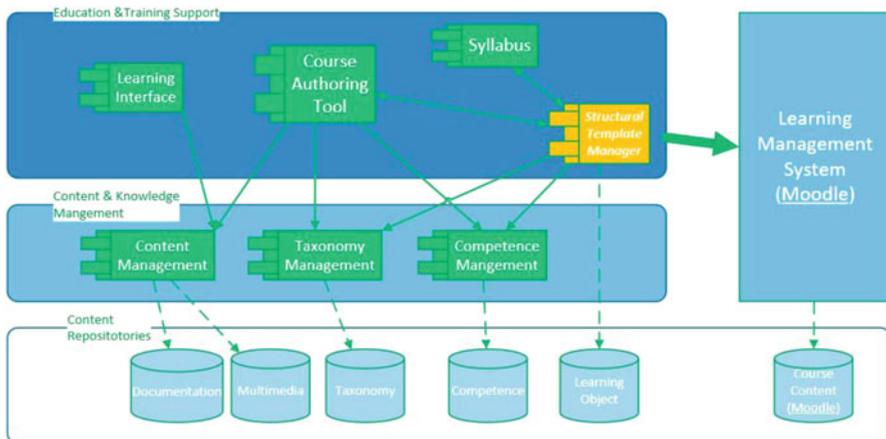


Fig. 4 New system architecture of DSTM in the KM-EP [7]

4.2 *Providing a RESTful Interface*

The Learning Design API is defined as a REST/RESTful interface with the following endpoints and provided by the KM-EP:

- GET: <URL>/webservice/learningdesign
This endpoint returns a list of all available DSTs.
- GET: <URL>/webservice/learningdesign/{id}
This endpoint returns the structure of a particular DST.
- GET: <URL>/webservice/competenceprofile
This endpoint returns a list of all available CPs.
- GET: <URL>/webservice/competenceprofile/{id}
This endpoint returns the content of a particular CP.
- GET: <URL>/webservice/competenceframework
This endpoint returns a list of all available CFs.
- GET: <URL>/webservice/competenceframework/{id}
This endpoint returns the content of a particular CF.

The exact return value's format will not be explained in detail here.

With this kind of access to the pedagogical structure and its contents (used CF and CPs), it is easier to make this access dynamically.

5 Initial Evaluation

There exist many different methods to perform an evaluation. For the DSTM we have chosen the method of a *Cognitive Walkthrough (CW)*. *In a CW domain experts put themselves into the role of imaginary users and evaluate the system from their perspective. The goal is to find the way of the least cognitive effort [6].*

To perform this CW, we have to introduce a scenario, which we will use. After we introduced this scenario, we will define an exemplary DST which we will transform into a normal Moodle course, and into a gamified Moodle course.

5.1 *Scenario for the Initial Evaluation*

The scenario we will use for the initial evaluation will be placed in the context of continuing vocational education. Within this context, we are e.g. faced the problem, that new employees have to be trained for their new work, and present employees have to be trained for e.g. new machines just like refreshing things they have to know, and necessary training courses, e.g. which have to be done by law.

For the CW we will use the case that a new plant will be built and the training should take place *before* the plant is already built. Especially we will define, how a

new employee will become a virtually guided tour through the new plant with some tasks, and how a new employee will become a virtual safety training.

5.2 Defining an Exemplary Didactical Structural Template

At first we will define the didactical structure of the described scenario using the DSTM. The didactical structure will roughly look as follows:

- local instruction with
 - general instruction,
 - automatic guided tour,
 - interactive guides tour,
 - security questions, and
 - final test
- security instructions with
 - general instruction,
 - instructions in industrial safety,
 - instructions in first aid,
 - instructions in fire prevention,
 - . . . , and
 - final test

To realize this, we will have to define a new DST. Therefore, we open the DSTM and click on the button “Create a Template” (Fig. 5).

After this, a new page with general information opens. After this information is filled in and saved, the content of the DST can be defined (Fig. 6).

All Structural Templates

Identifier	Title	Description	Author	Version	Actions
I2L	Immerse2Learn		Sergej Schachow	2.1	Install, Export, Edit, Delete
BE	Betriebliche Einweisung			2.2	Install, Export, Edit, Delete
TUDA-Int	Informatik Studium	Herzlich Willkommen am Fachbereich Informatik. Die Technische Universität Darmstadt ist die führende forschungsorientierte Universität im Großraum Rhein-Main-Neckar mit einem deutlichen Schwerpunkt auf Informatik und Ingenieurwissenschaften.	Minh Duc Hoang	1.0	Install, Export, Edit, Delete

Fig. 5 Overview of the existing DSTs [7]

Edit Structural Template: Betriebliche Einweisung

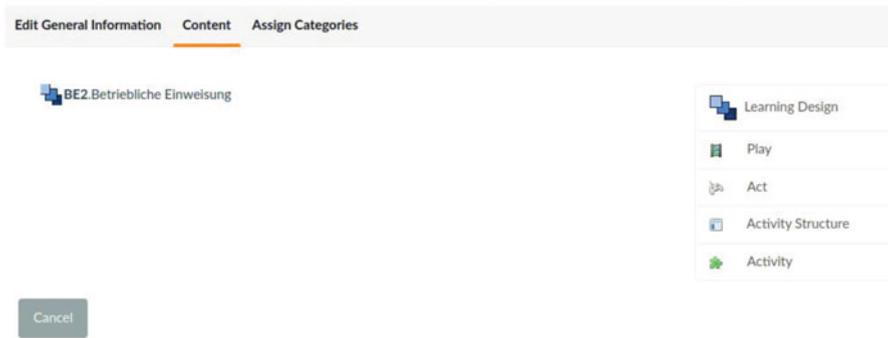
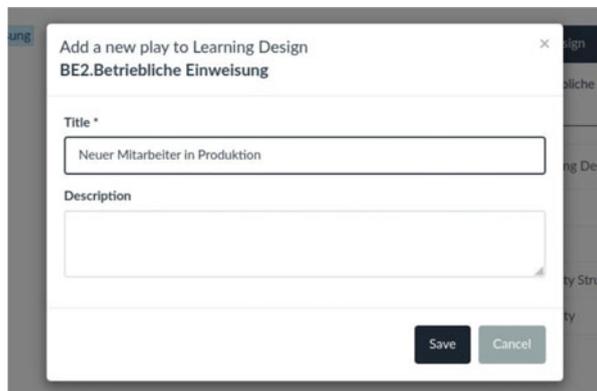


Fig. 6 Initial content of the new DST [7]

Fig. 7 Creating a new play—context menu [7]



Fig. 8 Creating a new play—defining the play [7]



By right-clicking on the elements, the sub-elements can be created. In this case, we can create a new play within the DST (Figs. 7 and 8):

Resulting in the following DST (Fig. 9):

Following this way, we can define the whole needed DST with the above given structure. This will lead to the following DST (Fig. 10):

Fig. 9 New DST with defined play [7]

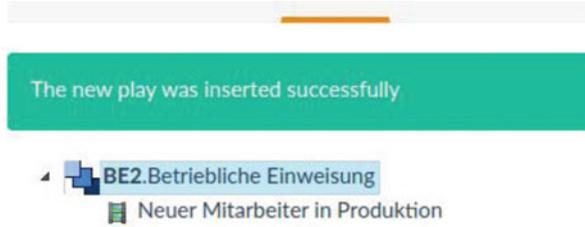
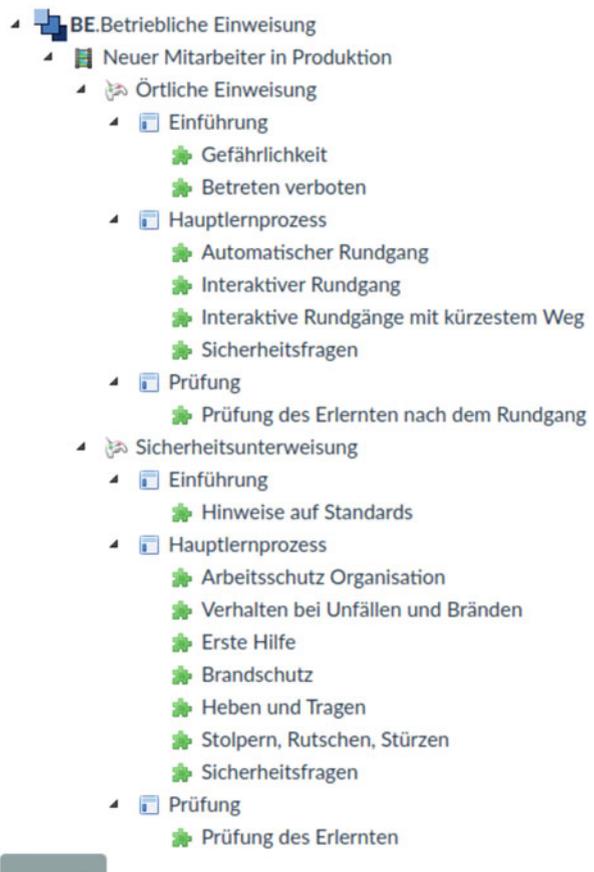


Fig. 10 Complete defined DST [7]



To define a prerequisite or a goal CP on an element of the DST, you just have to click on the corresponding menu item of the context menu (as in Fig. 7). Doing so, a new window will open and show the available CPs (Fig. 11):

The details of each element of the DST will be shown on the right side of the editor (Fig. 12).

For a description, how to define a CF and a CP, have a look in [9, 10].

Fig. 11 All available CPs [7]**Fig. 12** Details of a defined act in the final DST [7]**Fig. 13** Overview of the Moodle courses [7]

After defining the DST, we now can implement it into the Moodle courses in the next two sections.

5.3 *Creating a Moodle Course*

To implement a DST as a Moodle course, we have to go first to the overview in the DSTM (see Fig. 5). Then we have to click on the Install button next to the DST we want to implement as Moodle course.

After clicking on the Install Button, the DSTM will create a Moodle course with the necessary Moodle elements in the course (Figs. 13 and 14).

The Moodle course will initially be created with some default values for the Moodle elements. To define the needed learning content, we have to use the CAT within the KM-EP. For more details on how to use the CAT, have a look into [10].

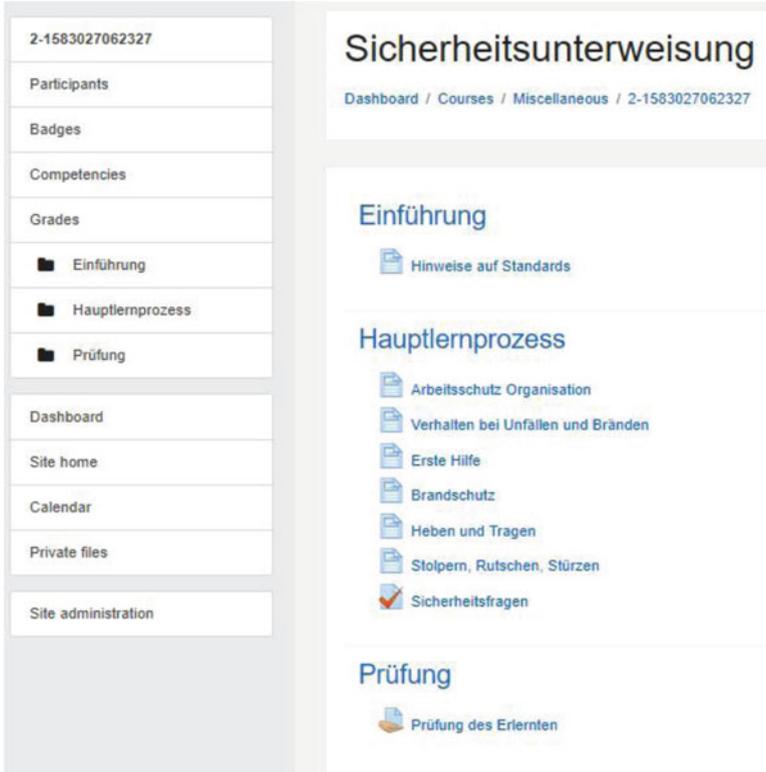


Fig. 14 Details of the implemented Moodle course [7]

5.4 Creating a Gamified Moodle Course

To implement a DST as a gamified Moodle course, at first we have to implement the DST as a Moodle course as described in the section before. The difference is, how the learning content of the course is defined.

To make the connection between an applied game and Moodle possible, we use the *Learning Tools Interoperability* (LTI, [2]). We already described in [13] how the LTI connection between Moodle and the applied game works. We will show exemplary, how the implemented applied game can be included within a Moodle course.

At first, the URL of the created applied game has to be added as software into the KM-EP. After the applied game is known within the KM-EP, it can be referenced by a Moodle course. Therefore, a learning element has to be defined as an asset which references the applied game with help of the CAT.

When a learning element with an underlying applied game is selected by the learners, the applied game will start, and the learning result will be transferred back to the LMS via LTI.

As a result of the initial evaluation is to say that it revealed minor errors, which have been resolved. As a further result, the user experience has been enhanced to make some processes for designing a DST simpler for the user.

6 Conclusions

In this paper, we had a deeper look into the conceptual design of our extension of the IMS-LD specification to provide competence information. We presented two different options to provide the structure of a DST and the contained CFs and CPs, which are to export the DST as file and to provide an API with which the DST can be requested. We also have shown a first initial evaluation of the described DSTM.

6.1 Future Work

Future work will be a further formal evaluation of the DSTs and the DSTM, especially in the way to use the DST as pedagogical structure for an applied game without connecting it to a Moodle course. The process of creating a Moodle course by installing it from a DST is not as good as it could be. To improve the installation process, it would be desirable to have a wizard, so that the learning content can already be filled in while installing the DST as a course. For the option, that a learner can switch between different implementations of a DST, it is necessary that within the DST every learning element, where it is possible to switch, has defined prerequisites and goal competences. Therefore, a competence profile for the learner has to exist, where the new goal competences can be added. Before a learner enters learning elements, there has to be a check of the prerequisites and the learner's competences, if the learner can take part (or exceed) the learning elements he wants. Important for switching between different implementations of a DST is that the learner does not try to switch between different DSTs.

Acknowledgments This publication has been produced in the context of the RAGE and Immerse2Learn projects. The projects have received funding from the European Union's Horizon 2020 Research and Innovation Action under grant agreement No 644187 and from European funds for regional development (EFRE). However, this paper reflects only the authors' views and the European Commission is not responsible for any use that may be made of the information it contains.

References

1. I.G.L. Consortium, Learning design specification (2003). <https://www.imsglobal.org/learningdesign/index.html>
2. I.G.L. Consortium, IMS learning tools interoperability (ITI) assignment and grade services specification (2019). <https://www.imsglobal.org/spec/lti-agrs/v2p0/>
3. D. Crockford, The javascript object notation (JSON) data interchange format (2017). <https://tools.ietf.org/pdf/rfc8259.pdf>
4. R.T. Fielding, Architectural Styles and the Design of Network-based Software Architectures. Ph.D. Thesis (2000). https://www.ics.uci.edu/~fielding/pubs/dissertation/fielding_dissertation_2up.pdf
5. GmbH: Ecosystem portal. <https://www.globit.com/products-services/educational-portal/>
6. M. Hegner, Methoden zur evaluation von software. Technical Report, Informationszentrum Sozialwissenschaften der Arbeitsgemeinschaft Sozialwissenschaftlicher Institute e.V. (ASI) (2003)
7. M.D. Hoang, Management of Structural Templates in association with Competence-Based Learning and the LMS Moodle. Master's Thesis, Technische Universität Darmstadt (2020)
8. Moodle: Moodle - open-source learning platform (2020). <https://moodle.org/>
9. M. Then, Supporting Qualifications-Based Learning (QBL) in a Higher Education Institution's IT-Infrastructure. Ph.D. Thesis, FernUniversität Hagen (2020). https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001608
10. B. Wallenborn, Entwicklung einer innovativen Autorenumgebung für die universitäre Fernlehre. Ph.D. Thesis, FernUniversität Hagen (2018). https://ub-deposit.fernuni-hagen.de/receive/mir_mods_00001428
11. M. Winterhagen, M. Then, B. Wallenbrn, M. Hemmje, Towards structural templates for learning management systems taking into account standardized qualifications. Technical Report, FernUniversität Hagen (2019). https://www.researchgate.net/publication/340601087_Towards_Structural_Templates_for_Learning_Management_Systems_taking_into_account_Standardized_Qualifications
12. M. Winterhagen, M.D. Hoang, H. Lersch, F. Fischman, M. Then, B. Wallenborn, D. Heutelbeck, M. Fuchs, M. Hemmje, Supporting structural templates for multiple learning systems with standardized qualifications, in *EDULEARN20 Proceedings* (2020)
13. M. Winterhagen, M. Salman, M. Then, B. Wallenborn, T. Neuber, D. Heutelbeck, M. Fuchs, M. Hemmje, LTI-connections between learning management systems and gaming platforms. *J. Inf. Technol. Res.* **13**, 47–62 (2020)

Enhancing Music Teachers' Cognition and Metacognition: Grassroots FD Project 2019 at Music College



Chiharu Nakanishi, Asako Motojima, and Chiaki Sawada

1 Introduction

There are three key bases behind the launch of the “Grassroots FD Project 2019.” First, FD (Professional Faculty Development) was mandated by the Ministry of Education, Culture, Sports, Science and Technology (MEXT) in Japan. Teachers who are in charge of the Teacher Training Course are required to have papers published on their specialty. Second, a music college has unique circumstances. Music performance teachers are characteristically different from other specialty teachers. In addition, the music lessons are closed, done privately between one teacher and one student, and have not been investigated. Third, there is a decline in the performance skills and academic ability of music college students. We would like to explain these backgrounds briefly.

In 2008, the Japanese School Education Law by the University Establishment Standards was revised, and FD became a “mandatory duty” (Article 25, Section 3). Since then, the university teacher professional training in Japan has been promoted as a part of FD. The survey in 2009 by MEXT shows that over 90% of higher education institutions work on FD as an activity of improvement. Some universities have adopted the class evaluation of the students only superficially, while other universities have offered systematic educational programs of teaching and learning which lead to real improvement of classes [1–3]. However, most institutions have no policy in place for the systematic engagement of FD or the professional development of academic staff enabling them to be comfortable with teaching and/or learning outcomes of training programs [4].

C. Nakanishi (✉) · A. Motojima · C. Sawada
Music Department, Kunitachi College of Music, Tokyo, Japan
e-mail: nakanishi.chiharu@kunitachi.ac.jp; motojima.asako@kunitachi.ac.jp;
sawada.chiaki@kunitachi.ac.jp

Japanese universities that have teacher training courses are urgently required to respond appropriately to the MEXT's accreditation. In order to obtain the accreditation of teacher training, teachers are required to have their papers published on the subject they teach. At universities, there are many types of teachers. Not all of them have experiences to write papers. Some are practitioners who have turned from the business world to the education world, and some are sports athletes and Olympic gold medalists. At a music college, some teachers are opera singers, and some are orchestra players. Though they are excellent music performers, they are not always skillful in teaching. As "the sensibilities are the most important in performance," the method of teaching is left to each teacher. Whereas in English teaching TESOL (Teaching English to Speakers of Other Languages) is established scientifically and academically, in music teaching, there is no special established teaching method. When the teachers have to face the students with lower ability of music performance, limited cognitive skills, and less motivation, the traditional teaching method based on the teacher's own experiences has become ineffective. Music teachers need opportunities to relearn their teaching to adjust to the situations and the students who were born after 2000. However, where we work for, Kunitachi College of Music, a private music college in Japan, the administrators adopted FD only by obligation. The mandatory FD loses its objective in solving the problems of teacher's paper achievement and improving teaching skills. It is necessary to provide opportunities that music teachers will train their professional skills and write their papers to meet the requirement of MEXT.

In this study, the three authors who are teachers of this music college will report an FD project from different viewpoints. Nakanishi, the first author, is an education major and the leader of this project. Motojima, the second author, is a vocal major, the project's subleader. Sawada, the third author, is a piano major. Nakanishi and Motojima are the representatives of this project, "Grassroots FD Project 2019." We are volunteer faculty developers of Kunitachi College of Music. We invited 12 teachers to join the project including Sawada. The role of Sawada in this paper was to review the project from a viewpoint of a participant to improve the future project with her younger generation colleagues. The organization chart of the project is shown (Fig. 1).

2 Theoretical Framework

2.1 Classroom Observations as FD

One of the common FD types is classroom observation. Classroom observation is the process of studying and analyzing classroom activities to scrutinize teaching strategies adopted by the teachers. Foster says that classroom observation helps teachers to test their personal theories on phenomena around them and refine social and psychological behavior of others and themselves [5]. Maingay also defines

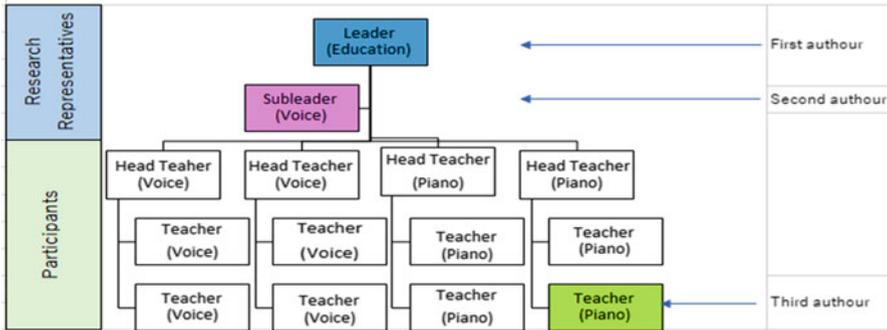


Fig. 1 Organization chart of Grassroots FD Project 2019

classroom observation as a reflective tool for the teachers to explore their own behavior [6]. To make sure that observation is purposeful and developmental, some guide such as what to observe and how to observe should be considered [7]. Allright [8] and Wajnryb [7] insist it is important to capture the events of the classroom accurately and objectively and not only to make a record of impressions. Recent researches emphasize the importance of observation procedures [9, 10]. They insist that highly structured observation has a clear focus and involves carefully prepared schedules, rating scales, and coding systems [9, 10].

2.2 Classroom Observation with Reflection: Cognitive and Metacognitive Questions

As Cohen, Manion, and Morrinson [9] and Mackey and Gass [10] mention, the clearer focus observation is required for effective FD. To structure the observation, the use of relevant question would be a key. Questions have been the essential component of many instructional methods including reviews, discussions, and problem solving. There are several types of classifying questions. One type refers to cognitive complexity. Questions may address various levels of cognition from recalling memory to creating something new. Bloom's taxonomy of educational objectives [11] and his former students' Anderson and Krathwohl's revised taxonomy [12] have been frequently used to classify questions. They have 6 cognitive process categories and 19 cognitive processes (Table 1). Each cognitive process category is hierarchical from the lowest order, *Remember* to the highest order, *Create*. *Remember* is the act of recalling information. The largest category of cognitive dimension is *Understand* which builds connections between the new knowledge and their prior knowledge. They are the acts of *interpret, exemplify, classify, summarize, infer, compare, and explain*. Questions of *Apply* involve using procedure to perform exercise or solve problems in familiar (*execute*) and unfamiliar situation (*implement*). Questions of

Table 2 A teacher asks eight questions to him/herself on the second phase of the ALACT model

	0. What is the context?	
	The teacher him/herself	The learner
Do	1. What did I do?	5. What did the learner do?
Think	2. What did I think?	6. What did the learner think?
Feel	3. What did I feel?	7. What did the learner feel?
Want	4. What did I want to do?	8. What did the learner want to do?

Adapted from Figure 8.2 Korthagen [15]

Analyze aim to break materials into its constituent parts (*differentiate*) and determine how the parts relate to one another or to an overall structure or purpose (*organize*), and determine a point of view (*attribute*). Questions of *Evaluate* determine whether a process or product has internal consistency (*check*) and whether a product has external consistency (*critique*). Questions of *Create* require to generate alternative hypothesis (*generate*), devise a procedure to accomplish a task (*plan*), or formalize a new product (*construct*).

2.3 Teacher's Reflection

Metacognition is the reflection about what people already know or is in the process of doing something [13]. Kolthagen emphasizes the importance of reflection for teachers. The process of reflection is shown in the ALACT model [14]. This model describes the ideal reflection process in five phases.

In the first phase, a teacher performs a class activity. In the second phase, "Looking back on the action," the teacher looks back on his or her own action, and answers eight questions about the student and him/herself, depending on the context at that time (see Table 2). By answering the eight questions, a teacher can begin to recognize the differences between what he/she thinks and what he/she feels and what he/she wants to do. Being aware of any inconsistency is an important step in "awareness" and "awareness" of the essential aspects of the third phase and will lead to an increase in options for actions in the fourth phase. In the fifth phase, the teacher chooses a new option and tries a new activity, by practicing this model cyclically, the fifth phase to the first phase.

2.4 The ICE Model: As an Analytical Tool

In Nakanishi and Motojima's former study of FD [16–20], the ICE model was applied to vocal learning (Table 3). Originally, the ICE model [21] represents the three phases of learning. (I) is an acronym of Ideas, which represents the building blocks of learning. (C) is an acronym of Connections, which represents establishing

Table 3 Summary of vocal learning based on the ICE model (by Motojima)

	Ideas (I) Techniques	Connections (C) Expressions	Extensions (E) Senses
Phases of leaning	To perform music notes accurately	To connect all elements of (I) To breathe life into music	Something “special” To reach someone’s heart
Elements of learning	Quality of voice/volume	Expressions of feelings	Imagination
	Quality of vibration	Will to express	Creation
	Pronunciation	Sound	Appreciation
Index of evaluation / analysis	Rhythm	Interpretation	Movement
	Phrasing	Analysis	Resonance
	Accuracy	Sympathy	Empathy

and articulating the relationships among components of Ideas. (E) is an acronym of Extensions, which represents learning is internalized and used in new ways. Table 3 shows the application of the ICE model to vocal learning by Motojima. (E) is something special to move people and touch someone’s heart. It is interpreted as “senses” in music learning. To reach someone’s heart at (E) phase, music techniques are essential, which are the base of learning music performance. We interpret Ideas (I) in music performance as techniques. The elements of techniques (I) are “quality of voice/volume, quality of vibration, pronunciation, rhythm, phrasing, and accuracy.” Learners are expected to learn these elements at (I) phase to perform the music notes accurately. Then, at the next phase of Connections (C), the elements of (I) are connected. The elements of Connections (C) are “expression of feelings, will to express, sound, interpretation, analysis, and sympathy.” The learners put together what they have learned at the (I) phase and breathe life into music. The framework of the ICE model helps the teachers to analyze the focus of music learning more clearly.

3 The Present Study

3.1 *The Purpose of the Study*

The purpose of the present study is to outline the development of the FD (university teacher professional training) project for music teachers and to examine how it can affect the teachers.

3.2 Development of the “Grassroots FD Project 2019”

The FD project at Kunitachi College of Music was started in 2015 by Nakanishi and Motojima, and our achievement was published in 2016 [22]. Since then, we have been continuing our study. The present study, “Grassroots FD Project 2019,” was started in 2017 and ended in March 2020 by publishing the book *FD at Music College, Grassroots FD project 2019, -Teacher's Awareness of Teaching-* [23]. The final goal of the project is to have music teachers, who are novice writers, write a paper based on their teaching to meet the requirement of the MEXT.

3.3 Activities, Learnings, and Worksheets

Connecting the above four theories mentioned in section “Theoretical Framework,” we developed the “Grassroots FD Project 2019.” The FD project has mainly six activities: observing open music lessons, answering worksheets with cognitive and metacognitive questions, discussing with colleagues, practicing and reflecting teaching, and writing a paper (Fig. 2). To support learning of the teachers, three worksheets were developed.

Figure 3 shows the link of activities, learnings, and worksheets. Worksheet 1 (Appendix 1) is named “Open lesson analysis sheet” which has mainly two parts. It is used while observing open lessons. The first part of the sheet is observing teachers evaluate the performance of a student and simulate teaching while an instructor is teaching. In the second part, the teachers analyze the open lessons by the ICE model. The teachers learn by observing and thinking including analyzing. Worksheet 2 (Appendix 2) is named “A teaching report: Improving lessons.” The focus of the teachers shifts from the open lesson to their own lessons. They list three ideas which they get from the instructor of the open lesson and apply them to their own lessons. They give experimental lessons with the ideas in their own teaching. They record their practice and reflect while and after teaching using the questions of Korthagen (Table 2). In the end, the teachers give a questionnaire or interview students what they think about the new teaching. With Worksheet 2, teachers learn by teaching and reflecting. Worksheet 3 is a template of a paper. By following the template, they can complete their paper. The teachers learn by reflecting and writing.

3.4 Enhancing Cognition and Metacognition

Figure 3 shows the process of the “Grassroots FD Project 2019” and explains the link between the project with cognitive process categories and cognitive processes [12].

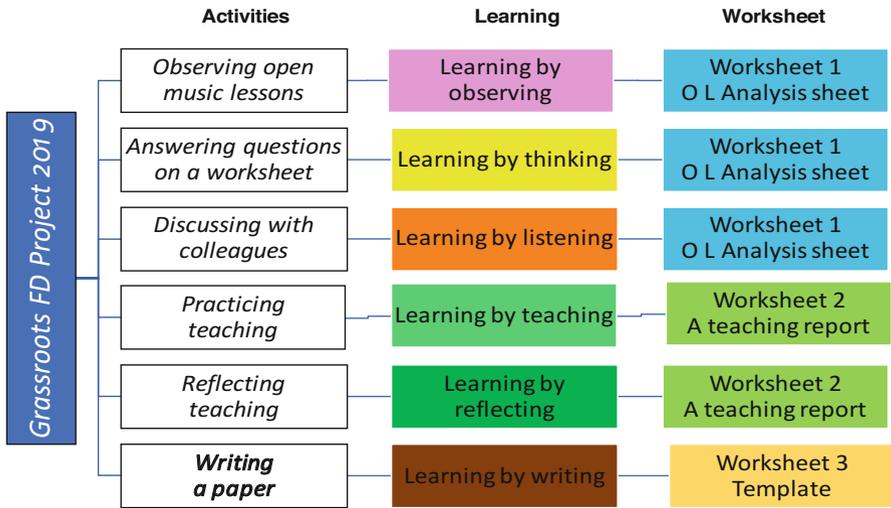


Fig. 2 Teacher’s activities, learnings, and worksheets

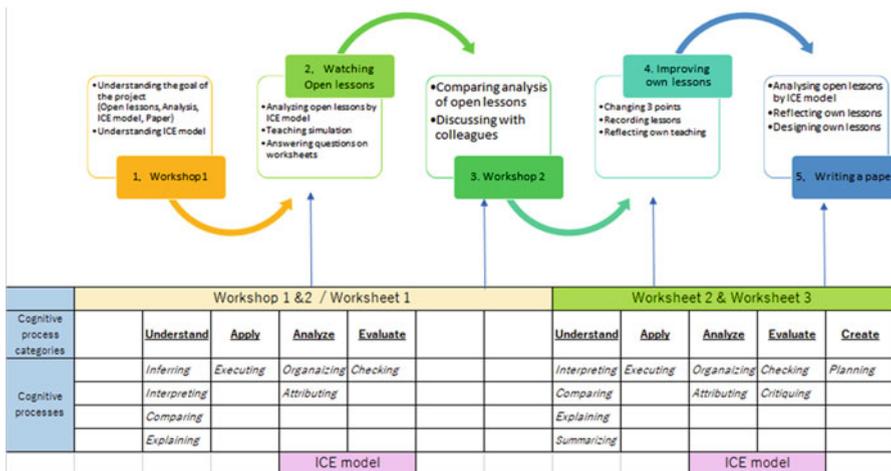


Fig. 3 Link of the project and cognitive process categories

In Workshop 1, the teachers are expected to understand the ICE model and to analyze music learning and teaching by this model. The teachers observe the open lesson and use Worksheet 1, “Open lesson analysis sheet” (Appendix 1), and answer the cognitive questions. By answering the questions, the four cognitive process categories (*Understand, Apply, Analyze, Evaluate*) and eight cognitive processes (*Inferring, Interpreting, Comparing, Explaining, Executing, Organizing, Attributing, Checking*) are promoted. When they evaluate students’ performance and analyze teaching of instructors, they use the framework of the ICE model (Table 3).

In Workshop 2, the teachers report their results of analysis and share them with their colleagues based on “Open lesson analysis sheet.” After this workshop, they list three ideas to improve their own lessons. Then, the teachers apply these three points which they list to their own lessons (Cognitive process category, *Apply*; Cognitive process, *Executing*). They are expected to follow Worksheet 2, “A teaching report: Lesson improvement” (Appendix 2), and recorded their lessons with a voice recorder. The teachers give a questionnaire about their lessons and interview students about what they think about these new ideas of teaching.

Using Worksheet 2 and Worksheet 3, “A template” (Appendix 3), the teachers are expected to reflect on their teaching and write their findings in about four to nine pages. They are expected to reflect more deeply and criticize their teaching by checking recorded lessons and students' voices. After reflection, some teachers design a new lesson plan. The cognitive process categories (*Understand*, *Apply*, *Analyze*, *Evaluate*, and *Create*) are promoted by answering the questions. The cognitive processes are *interpreting*, *comparing*, *explaining*, *summarizing*, *executing*, *organizing*, *attributing*, *checking*, *critiquing*, and *planning*.

4 Results of the “Grassroots FD Project 2019”

4.1 Participants

In the FD project, 13 music teachers of Kunitachi College of Music participated. They are seven vocal teachers and six piano teachers (Fig. 1). The cooperators of the project were 2 instructors of open lessons (voice instructor, Prof. Chiyoe Sho; piano instructor, Prof. Pascal Devoyond), 6 students of the open lessons (voice, $n = 4$; piano, $n = 2$), and 90 students (voice, $n = 53$; piano, $n = 37$) who took lessons of 13 music teachers.

4.2 Targets of the Lesson Improvements

The 13 music teachers made experimental lessons from July to December 2019, after observing the vocal open lesson (July 11, 2019) and the piano open lesson (October 3, 2019). The target lessons were the private lessons for the vocal and piano students and the group lesson for nonvocal and piano students. The total number of these experimental lessons was 398.

4.3 Teachers’ Learning from the Grassroots FD Project 2019

After participating in this project, the 13 teachers each finished writing about a four- to nine-page paper, “A teaching report: Lesson improvement,” to meet the requirement of MEXT. What did they learn from this project? We evaluated the project from their writings.

Observing open music lessons Observing another instructor’s teaching at open music lessons stimulated teachers and made them think about teaching more. Some teachers said open lessons were the only chances to see other teacher’s lessons.

Answering questions on a worksheet Five teachers commented on the effects of the questions of a worksheet. One mentioned that to follow the questions, she could observe open lessons not from a viewpoint of a performer but from that of a teacher. She also stated that the questions covered almost all aspects of teaching to check. Among the questions, to analyze open lessons was the main objective of the FD project. Motojima mentioned that the ICE model helped teachers avoid criticizing the instructor personally but focusing on instructor’s teaching itself. The other teacher said the ICE model helped to visualize the music performance and was easy to understand.

Discussing Discussing the ways of teaching of the open lessons with their colleagues seemed to play an important role for all teachers. They said that they had never exchanged their opinions about teaching before the project. They could learn from other teachers and gained new ideas about teaching.

Applying three ideas of teaching to teachers’ own lessons The teachers list the three ideas of teaching which they gained from the open lessons and applied to their own lessons. The ideas are listed in Table 4. There are two common characteristics in these ideas. First, many teachers would like to communicate with students more. They would like to know what their students think. Second, they would like to ask cognitive questions to have students think more.

The teachers summarized what they learned by giving experimental lessons with three new ideas and practicing, in about 50 words. This summary showed their reflection on the project clearly. Analyzing the reflection revealed that the focus of 10 out of 13 teachers was on their metacognition. They looked at their teaching

Table 4 Excerpt from “Three points to improve teachers’ own teaching”

Vocal teachers	I’d praise students. I’ll give positive feedback.
	I’ll ask studnets what is difficult for them.
	I’ll narrow the focus of my instruction.
Piano teachers	I foster students’ mind of trial and error.
	I’ll lead what studnets want to do and can do in their performance.
	I develop students’ ability to listen.

critically and objectively and reflected their teaching from the standpoint of a third party. Regardless of the teachers' specialties of either voice or piano, there were many descriptions of teaching in meta-state. The examples are "I recognize that I myself want to teach so as my students would understand me easily." (vocal teacher), "By giving my students enough time to digest the essential vocalization, they produced better voice and felt the effect." (vocal teacher), and "I deeply considered the process of unconsciousness and consciousness until the students complete the piece of the music." (piano teacher). Six vocal teachers focused on their way of expressing, but only one piano teacher mentioned about it. One of the vocal teachers wrote: "Asking the students many cognitive questions improves their thinking. The conversation between the students and myself increased, and I was able to draw out the students' willingness to perform more than before." Five out of six piano teachers focused on analytical skills, while vocal teachers mentioned only one in seven. For example, one teacher wrote: "My piano teaching shifted from 'giving (knowledge)' to 'have students feel music, think about music, and play the piano by applying what they feel and think.'" The description of this piano teacher is analytical and shows three steps of "feel," "think," and "apply."

They affirmed that workshops and worksheets helped them to write, but some said they were not helpful on which topic they should pick up or develop. One teacher who was a weak writer said that this writing experience helped her understand weak piano students. She got to know how students felt when they didn't understand how to improve with the piano.

4.4 Review of the "Grassroots FD Project 2019" by Chiaki Sawada

In this section, one of the piano teachers, Chiaki Sawada, reviews the project from the viewpoint of a participant.

In Workshop 1, my colleagues and I found the ICE model was hard to understand. We were not used to analyze music performance teaching and categorize it into three phases. However, feeling the difficulty of understanding the ICE model led us to think more about music performance and its teaching. After thinking over carefully about the ICE model, we reached our own interpretation of using the three phases of the ICE model to music performance. I thought music itself is the same as the ICE model in some aspects. Both music and the ICE model have their own theories which are inherently invisible and sensual. Moreover, both of them are interpreted, analyzed, and expressed in several ways. The method of interpretation, analysis, and expression of music and the ICE model are not always limited to one. I think music performance and teaching music performance are basically inexplicable. However, analyzing music performance and its teaching by the analytical tool of the ICE model helped us to see what they are more clearly.

The advantage of analysis by the ICE model was that it was a flexible framework that could classify music performance and its teaching into three broad-based phases. Teachers can interpret and apply their own concepts of music into the ICE model. There are a variety of interpretation depending on their teacher's experiences and ideas. Because of these reasons, I thought the ICE model was suitable for analyzing music performance and its teaching.

To tell the truth, Prof. Nakanishi and Prof. Motojima's explanation of the ICE model in Workshop 1 was confusing. After a briefing of the ICE model, several cases of using the ICE model were introduced. However, this detailed explanation blurred our understanding. The concept of the ICE model and the idea of analyzing music performance from this framework were new to us. The focus of explanation should be narrowed down. However, the handouts given at this time were well organized and very easy to understand. By reading the materials at home, I was able to understand the whole picture of the project.

Though we depended on Worksheet 3, a template, I found the points which should be improved. First, in the first section, each teacher wrote very differently. I suppose the template should give us more detailed explanation on what we should write. Second, I found some discrepancies in Worksheet 2 and Worksheet 3. In Worksheet 2, there was a place to write about the process of what we did in our lesson, but this part was hardly covered in Worksheet 3. I thought that if the process of analysis was also explained in Worksheet 3, the readers might understand how the teachers chose their three points to be improved. Then, it would make our paper even more interesting. Third, since many music teachers are unfamiliar with writing a paper, a brief writing workshop might be helpful. I simply didn't know how to title a paper, label figures and tables, and even write headings in paragraphs.

The book of "Grassroots FD Project 2019" was arranged and edited by Prof. Nakanishi in the form of "academic paper" to meet the requirement of MEXT. When we, the teachers, were writing a part of this book, we thought we were writing an academic paper. But I found it was just a practical report. I recognized this, when I read some chapters written by Prof. Nakanishi. We, the teachers, wrote the important parts of the book, but from the viewpoint of the whole research, what the teachers wrote was not academic. I think that the teachers could analyze teaching by using ICE model using their professional skills as music experts and university teachers. However, regarding the academic paper, the teachers were novice writers. In this sense, Grassroots FD Project 2019 made me aware and realize the differences of a report and an academic paper. It was hidden but genuine FD.

5 Conclusion

Nakanishi and Motojima started the FD project so that music teachers who are novice writers would write the paper to meet MEXT requirements. "A teaching report: Improving lessons" written by 13 teachers did not present any unique perspective or conclusion, and it was not academic. However, it made a difference

from a report which presents only feelings and impressions. Based on teachers' observation and teaching experiences, they wrote with objective and analytical view using the ICE model and Bloom's revised taxonomy. As the leaders of this project, we thought that the goal of the project was not to "teach the teachers," but to provide an opportunity "to raise awareness of their teaching." To that end, we developed three worksheets and organized workshops and even provided individual mentoring. The worksheets with various questions seemed to activate the teacher's cognition and encouraged their metacognition: teaching simulation, comparing teaching of the instructor with their own teaching, analyzing lessons by the ICE model, applying new ideas to their own teaching, recording lessons, reflecting, designing new lessons, etc. Among them, the most useful one for teachers seems to be to analyze music performance by the ICE model. Seeing lessons from the framework of the ICE model is like looking at the world from haiku (a short Japanese poem) 5-7-5 framework. Seeing from the new framework creates awareness. Teachers seem to connect analysis by the ICE model and writing a paper successfully. The ICE model gives different teaching views to the teachers who have been teaching for many years. The advantage of the ICE model was that the interpretation of the performance could be different for each individual teacher. Therefore, it was easy to accept as a framework for analyzing music.

In reflection of teaching, why did piano teachers focus on "analyzing" and vocal teachers on "expressing?" As the number of teachers who participated in the project is small, it is not possible to generalize their features of specialty. The following assumption can be made. Piano teachers do their best to analyze a piece of music in their minds before expressing, while vocal teachers use their "mouth" and "words" before analyzing. In other words, piano teachers may give priority to analysis not only in performance but also in instruction, and vocal teachers may tend to give priority to expression in any situation.

It is said that cognitive skills are developed not in the same situation or subject for everyone, but in the area in which each one is most interested and strong. People may want to know more or be better in areas where they are interested or excel, and they may have a sharpened sense. However, in fields that you are not interested in or unfamiliar with, you cannot think of anything critical. When teachers try to focus on one thing, they may look at the "area" that they are working on most sincerely.

One of the limits of this study is that the effect of FD was not measured. It is conducted only by teachers' surveys and interviews with students. The FD project should have looked at how the students perceived the lesson improvements. Second, a follow-up survey of how the teachers' awareness and the way of teaching have changed after participating in the FD project will also be necessary. Third, in this study, the review was done by a single teacher. It might not show a complete picture of the project. In our future project, we should provide questionnaires and interviews for all of the students and a follow-up survey and review for all teachers.

Acknowledgments This work was supported by JSPS KAKENHI Grant Number JP 19K03035.

A.1 Appendix

A.1.1 *Appendix (1) Excerpt of Worksheet 1: Open Lesson Analysis Sheet*

[Before] Listen to the first performance of the student/before the lecturer comments

1. Evaluate the student's first performance using the frameworks of the ICE model with numbers (1–5). Ideas (), Connections (), Extensions ()

If you were a lecturer, what would you like to focus on and teach to the first performance of the student? Explain the analysis of the above figures of the ICE model.

[While: During the open lesson]

2. Take a note of the lecturer's instruction.

3. Could the student understand the lecturer's instruction? If you were the lecturer, what and how would you like to bridge the gap?

4. Compare and contrast the lecturer's teaching and your teaching. From the lecturer's teaching, what do you want to apply to your own lesson? What is unacceptable?

[After] After the open lesson:

5. Look at your own note and analyze the student performance and the lecturer's instruction from the viewpoint of the ICE model. Put an acronym for (I)/(C)/(E) after each sentence.

A.1.2 *Appendix (2) Excerpt of Worksheet 2: A Teaching Report: Improving Lessons*

1. What do you think the lecturer is focusing on her/his open lesson? Indicate the focus using the ICE model with numbers (1–5).

2. Explain the analysis of the above figures of the ICE model, including specific examples (such as the lecturer's verbal explanation, behavior, and student responses). What and how did the lecturer teach? What did the lecturer particularly try to convey? (I)/(C)/(E)

3. Compare and contrast your teaching with the lecturer's teaching. Write down three things that you want to incorporate in your lesson, and write down the reasons. Analyze where the focus is in (I), (C), and (E) and note in the right column.

4. Make a detailed record of the practice of the three things you actually tried in your lesson (target student, duration, appearance/change of student, comment). Analyze the instruction in (I), (C), and (E) and write it in the right column.

5. Future tasks

A.1.3 *Appendix (3) Excerpt of Worksheet 3: A Template of a Paper*

Title (Think about the characteristics of the instructor's lesson and your improvement points)

1. Open lessons

Summarize what you think impressive (e.g., the performance and the remarks made by the instructor). Explain what you think about the instructor's open lessons, what is different from your own lessons, what inspired you, etc.

2. Lesson analysis

Analysis by the ICE model should be shown as number and figure.

3. A teaching report

References

1. H. Matsukawa, I. Murayama, N. Sakai, Y. Ishigami, Practice study of improving teaching methods through mutual classroom design and practice of the collaborative analysis method of learning processes in classrooms for enhancing capability to improve lessons. *J. Center Educ. Res. Teach. Dev.* (7), 511–558. (in Japanese) (2009)
2. F. Kumi, Y. Oishi, T. Hirota, K. Nagase, Creating the teaching materials for improvement education in a university as faculty development focusing on methods of active learning. *J. Yamaguchi Prefect. Univ.* **110**, 77–84. (in Japanese) (2017)
3. Y. Masayuki, Practice study of improving teaching methods through mutual classroom visitations and self-observations: A case study at a professional graduate school. *J. Japan Prof. Sch. Educ.* (8), 11–116. (in Japanese) (2015)
4. K. Kato, University teacher training in Japan. *Revista de Docencia Unifersitaira* **11**(3) Octubre-Diciembre, 53–63 (2013)
5. P. Foster, *Observing Schools, a Methodological Guide* (Paul Chapman Publish Ltd., London, 1996)
6. P. Maingay, Observation for training, development or assessment? in *Explorations in Teacher Training: Problems and Issues*, ed. by T. Duff, (1988)
7. R. Wajnryb, *Classroom Observation Tasks: A Research Book for Language Teachers and Trainers* (Cambridge University Press, Cambridge, 1992)
8. D. Allright, *Observation in the Language Classroom* (Longman, New York, 1988)
9. L. Cohen, L. Manion, K. Morrinson, *Research Methods in Education* (Routledge, Canada, 2000)
10. A. Mackey, S.M. Gass, *Second Language Research: Methodology and Design* (Lawrence Erlbaum Associates, Inc., Mahwah, 2005)
11. B.S. Bloom (ed.), M.D. Engelhart, E.J. Furst, W.H. Hill, D.R. Krathwohl, *Taxonomy of Educational Objectives, Handbook I: The Cognitive Domain*. New York: David McKay Co Inc. (1956)
12. L.W. Anderson, D.R. Krathwohl (eds.), *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom's Taxonomy of Educational Objectives*. (Complete Edition) New York: Longman (2001)
13. F. Smith, *Understanding Reading: A Psycholinguistic Analysis of Reading and Learning to Read*, 6th edn. (Mahwah, 2004)
14. F. Kolthagen, Professional development, learning and working in flow. <https://korthagen.nl/en/focus-areas/professional-development- teachers/> (2 Mar 2020)
15. F. Kolthagen (ed.) J. Kessels, B. Koster, B. Lagerwerf, T. Wubbels. *Linking Practice and Theory: The Pedagogy of Realistic Teacher Education* (Lawrence Erlbaum Associates, Mahwah, 2001)
16. C. Nakanishi, A. Motojima, Framework for designing and reflecting vocal lessons at Music College, an attempt to use ICE model, in *The 2018th Hong Kong International Conference on Education, Psychology and Society*, Royal Plaza Hotel, (Hong Kong), Proceedings, pp. 88–95, 2018
17. C. Nakanishi, A. Motojima, Using ICE model in vocal education. *J. Kunitachi Coll. Music* **52**, 231–239 (2018)
18. C. Nakanishi, A. Motojima, Faculty development program model based on open music lessons, in *2019 ISBASS International Symposium on Business and Social Sciences*, Grand Victoria Hotel (Taipei), Proceedings, pp. 25–36. 2019
19. C. Nakanishi, A. Motojima, S. Horie, Y. Shindo, N. Sakaguchi, K. Yamamoto, A study of FD based on open lessons at Music College -using ICE model approach-, *J. Kunitachi Coll. Music*, **53**(1), 139–150 (2019)
20. C. Nakanishi, A. Motojima, FD (Faculty Development) project for music teachers -using open music lessons-, *J. Kunitachi Coll. Music*, **54**, 113–124 (2020)

21. S.F. Young, R.J. Wilson, *Assessment and Learning: The ICE Approach* (Portage and Main Press, Winnipeg, 2000)
22. C. Nakanishi (ed.), *Improving Cognitive Skills at a Music College*. (in Japanese) (Asukai Press, 2016)
23. Chiharu Nakanishi, Asako Motojima (eds). *FD at Music College "Grassroots FD Project 2019" -Raising Teacher's Awareness of Teaching-* (in Japanese) (Asukai Press, 2020)

Scalable Undergraduate Cybersecurity Curriculum Through Auto-graded E-Learning Labs



Aspen Olmsted

1 Introduction

Demand for cybersecurity workers is estimated to increase 350% between 2013 and 2021 [1]. In response, large universities have created online cybersecurity graduate programs with reduced tuition to attract adult learners. New York University (NYU) established a scholarship program called Cyber Fellows that provides a 75% scholarship to all US eligible workers [2]. Georgia Institute of Technology (Georgia Tech) has created an online MS degree that can be earned at a cost of fewer than 10,000 dollars [3]. Both of these programs are designed to scale to thousands of students. Fisher College [4] is a small minority-serving private liberal arts college located in downtown Boston, MA. At Fisher College, we have designed an undergraduate program designed to serve our students online with not only scalability but also integrity.

Computer science, like many of the sciences, devotes a great deal of the students' time in the learning to hands-on lab activities. Cybersecurity is a sub-discipline within computer science that combines core application, programming, and database courses with upper-level information technology, computer science, and cybersecurity course. In programming courses, these labs take the form of the students writing application code in the language of the course. In database courses, the students are often submitting SQL queries in response to question prompts. In information technology and cybersecurity courses, the labs are often steps taken on real systems to configure systems or to eliminate vulnerabilities.

In face-to-face classes, instructors often use reverse classrooms so the students can have hands-on time with the instructor or a teaching assistant (TA), so when

A. Olmsted (✉)

Fisher College, Department of Computer Science, Boston, MA, USA

e-mail: aolmsted@fisher.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_59

825

they get stuck, they can get started again quickly without a long duration between submissions. The students watch lectures, read and take quizzes at home, and work on the labs in person to facilitate the just-in-time assistance. We show that students who learn with uninterrupted time do better in the completion of the labs.

For students in online classes, there is often a long time between a question and submission and the response or feedback that allows the student to continue learning in the lab. Online auto-graded systems help in the sense that the student will get some feedback immediately, but the student may have to wait for online office hours or a response to a forum post to continue with work. There is also a problem of ensuring integrity that the student submitting the work is the student who did the activities in the lab.

In this paper, we describe a technique we use in developing auto-graders that allows the student to receive feedback quicker while improving the integrity that the submitter is the author of the lab. The feedback comes in the form of auto-grader unit test results, along with allowing for peer discussions around the assignments. The number and quality of peer discussions increased because, in some cases, each student has a unique derivate of the lab that the students are completing. So instead of trying to stop student peer communication about lab solutions, we can encourage student sharing. The peer-to-peer student sharing has increased the students' understanding of the lab.

The organization of the paper is as follows. Section 2 describes the related work and the limitations of current methods. In Sect. 3, we describe the elements in the secBIML programming language. Section 4 explains the auto-graders we developed for our database courses. Section 5 describes how we developed our auto-graders for programming courses. Section 6 drills into the way we auto-grade for information technology courses. Section 7 investigates the way we build auto-graders for upper-level computer science courses. In Sect. 8, we drill into the auto-graders in our cybersecurity upper-level courses. Section 9 looks at our research questions and preliminary empirical data. We conclude in Sect. 10 and discuss future work.

2 Related Work

Jeffrey Ulman [2] developed an e-learning system with derivate questions called Gradiance Online Accelerated Learning (GOAL). GOAL provided quizzes and labs for several core computer science topics, including operating systems, database design, compiler design, and computer science theory. Each course was linked to a textbook with several quizzes per chapter and sometimes a few labs. The examinations were composed of questions with separate pools of correct and incorrect answers. When a student takes an exam, they are presented with a multiple-choice quiz where one right answer and several wrong answers are displayed for the student to choose the correct answer. A standard configuration of the system was four correct answers and eight incorrect answers. This configuration yield 224

derivate questions per each original item in the quiz. We have used GOAL over the years in database courses and found the derivative quiz questions allowed the students to discuss the exam without giving away the answer. The derivatives also will enable an instructor to answer a single derivate of an item in an online lecture. Unfortunately, GOAL only proved derivatives in the quizzes and not in the labs. The labs provided by GOAL were auto-graded, giving students immediate feedback, but since all students were working on the same labs at home, it was hard to stop answer sharing. Our work here supplements the job done in GOAL by providing not only automated grading of labs but also derivate questions per student (Fig. 1).

McGraw Hill [3] produces a commercial e-learning product called SIMnet. SIMnet's ambition is to teach students the skills required in the utilization of the Microsoft Office suite. SIMnet provides auto-graded labs that grade the students' submissions of database, spreadsheet, presentation, and word processing software. Students can learn the skills through online lessons that present the tasks in both reading and video format. SIMnet does protect the integrity of each student's work by inserting a unique signature into the starter file that the students download. If a student tries to upload a file with a different student's signature, the system catches the integrity violation. Either the upload is rejected, or the instructor is notified, depending on the lab configuration. Unfortunately, the labs that the students perform are not differentiated between students, so nothing stops one student from copying the work in the other students' files. Our work here improves on the integrity of the students' submission by deriving a different problem per student so they cannot just copy the other student's work.

Gradescope [4] sells a commercial e-learning product that allows instructors to scan student paper-based assignments. The grading of the paper-based assignments can then be automated through the e-learning system. The scanning feature has driven many mathematics and science departments in universities to adopt the system. A relatively unknown function of the system is the auto-graded programming framework. Gradescope designed a system that allows a student to upload a file for an assignment. The system then spins up a Docker [5] Linux session that is configured for the task. Test cases are developed in the auto-grader configuration with specific grading weights assigned for each test. We utilize this auto-grading environment for our derivative-based SQL, Python, and C#-based labs.

3 Database Auto-Grader

The auto-grader we developed for the database courses creates a Docker environment with a MySQL database running on a Linux environment. The students upload their query with a specific name: query.sql. The auto-grader then reads the metadata about the assignment to determine the assignment name and performs between three and five tests. Each test is weighted at two points each.

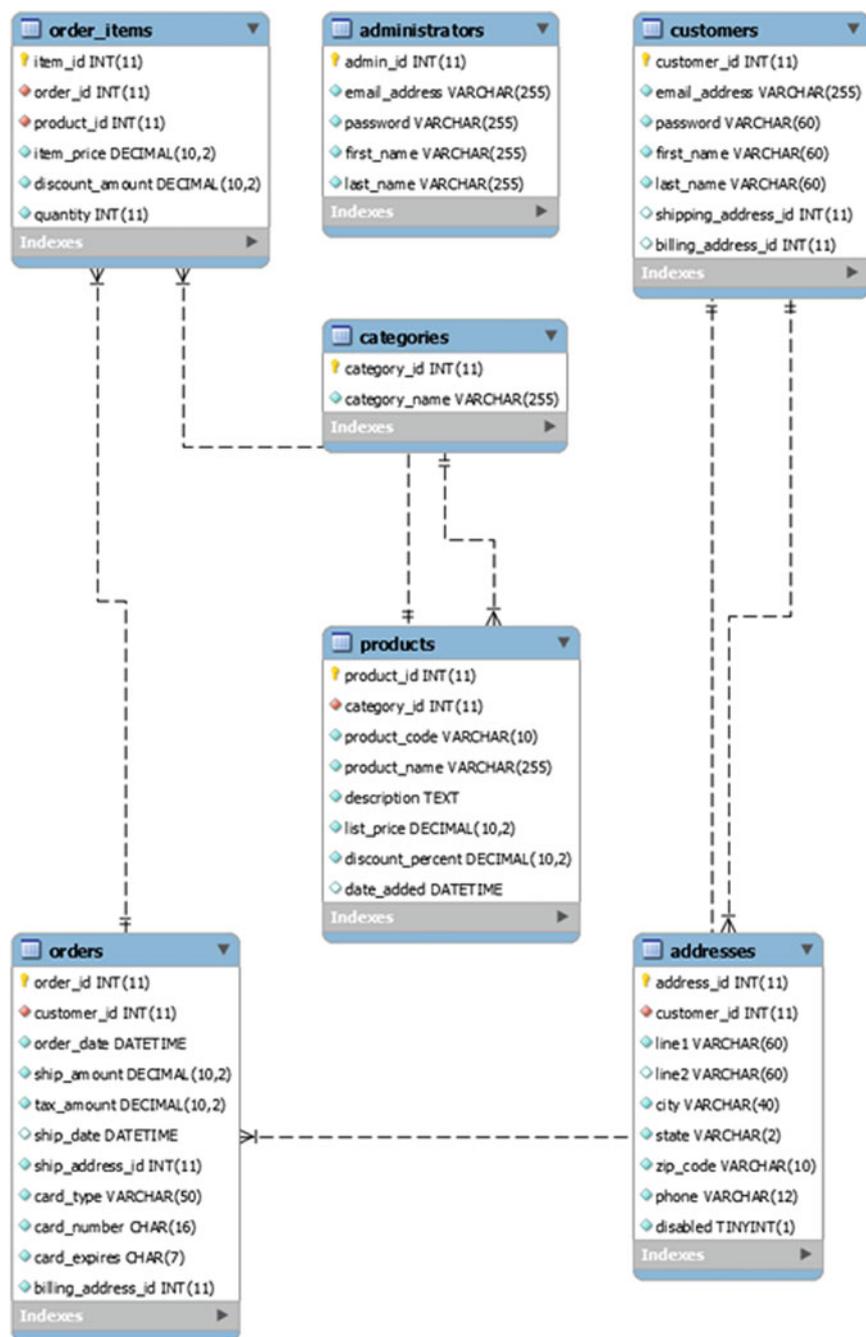


Fig. 1 Student lab ER diagram

Table 1 Sample value test

Field	Value
Name	Assignment 1
Value 1	Product_code
Value 2	Prodcut_name
Value 3	List_price
Order	List_price
Derivative	Random row

The test is actually stored in the MySQL database that is installed in the Docker session. The database connected has both the test information for the auto-graders and the same data provided to the students for their lab. There are two types of unit tests:

- Value tests
- Existence tests

For the value tests, each assignment has a row in the `value_unit_test` table. Table 1 shows a sample assignment row for a query that returns a specific product record. The primary key for the `value_unit_test` table is the name of the assignment. The `value_unit_test` also contains three value tests, along with an order test. The last value in the `value_unit_test` is the derivate method. Currently, supported derivate methods are:

- Random row – In this derivate method, the student's login is converted to a unique number between 1 and the maximum assignment number. The unique number comes from an order the student ID comes in the roster. The system will read the specific record in the order by value from the database to prompt the student to return that record.
- Random range – This is similar to random row but asks the students to return a tuple between a start and end value. The start value is the same as the random row value, and the end value is the fifth value after that record

Figure 3 shows the entity-relationship (ER) diagram for the student lab database. An assignment description website was built to display the question values that match the auto-grader tests for the student logged in. The auto-grader will score the student on ten possible points distribute across five tests:

1. Did the query execute?
2. Did the values match for the first value?
3. Did the values match for the second value?
4. Did the values match for the third value?
5. Did the order match?

Existence tests are similar to value tests, except they are used to grade queries that mutate the database such as insert, update, and delete statements along with queries that create views, functions, stored procedures, and triggers. In the case of existence

tests, the auto-grader will score the student on six possible points distributed across five tests:

1. Did the query execute?
2. Did test 1 pass?
3. Did test 2 pass?

Table 2 shows an example of an entry for the `existence_unit_test` table. The case is from an assignment where the student needs to write a query to create a new index. The test types either exist or do not exist. The test will either pass or fail if there is a value returned from the query. For an existing unit test type, data should be returned for a success. For not existing unit test type, no data should be returned for a successful test. Instead of a derivative based on a specific record as we used in the value unit tests, replacement variables are used to change the queries. Table 3 shows the available replacement variables. The variables allow names to be based on the user logged in, tables, and columns to be different for each student and literal string and numbers to be randomized.

Table 2 Sample existence test

Field	Value
Name	Assignment 2
Test 1	Show index from @RandomTable where key_name = @login_orders_ix
Test 1 type	Exists
Test 2	Show index from @RandomTable where key_name = @login_orders_ix and column_name = '@RandomColumn
Test 2 type	Exists

Table 3 Replacement variables

Variable	Meaning
@login	The user code for the logged in user
@RandomTable	A random table from the students sample database. This variable can be suffixed with a number between 1 and 9
@RandomColumn	A random column from the random table selected in the variable above. This variable can be suffixed with a number between 1 and 9
@RandomWord	A random word from the dictionary
@RandomInt	A random possitive integer

4 Programming Auto-Grader

We had previously developed a set of auto-graded foundational programming assignments in Python, Java, and CSharp for students in a first programming class. Unfortunately, many of these assignments did not lend themselves to derivatives that required different solutions per student. In our first attempt, we randomized the variables in the test cases to ensure students were not hard coding the output to match the input tests. To illustrate the challenge, we will itemize the labs below:

- Labs to practice programming expressions:
- Hello World – In this assignment, the student just outputs the words “Hello World.”
- Coin Counter – In this assignment, the student would be sent input variables for the number of quarters, dimes, nickels, and pennies and would output the total in dollars and cents.
- Coin Converter – In this assignment, the students would be given dollars and cents, and they would output the minimum number of coins by denomination.
- BMI Metric – In this assignment, the student would sent input of weight in kilograms and height in meters, and they would output the BMI.
- BMI Imperial - In this assignment, the student would be sent input of weight in pounds and height in inches, and they would output the BMI. The students would need to convert the imperial measurements to metric before calculating the BMI.
- BMI Metric with Status – This assignment is a modification of the earlier assignment and adds a decision branch to display a status of underweight, normal, overweight, or obese. The students have not learned decision branching yet, so the expectation is they will use modular division for this problem.
- Labs to practice programming iteration and decision branching:
- Cash Register – This assignment allows multiple inputs of item prices along with a club discount card and tax rate. The student outputs the base price, price after discount, and total price.
- Call Cost – This assignment provides the students with a rate table based on the day of the week and time of day. Input is sent with the day, time, and duration of the call and the students’ outputs the total cost for the request.
- Even Numbers – This assignment has the student output a certain number of event numbers based on the number input.
- Fibonacci – This assignment has the student produce the first n Fibonacci numbers. The number n is sent as input to the program.
- Labs to practice programming string operations:
- String Splitter – This assignment tests the student’s ability to divide up an odd length input string into middle character, string up to the middle character, starting after the middle character
- Character Type – This assignment has the student read a character of input and classify it into a lowercase letter, uppercase letter, digit, or non-alphanumeric character.
- Labs to practice programming functions:

- Leap Year Function – This assignment has the student write a function that takes a parameter and return true if the year is a leap year
- First Word Function – This assignment sends a sentence as a parameter to a function the student writes, and the student returns the first word of the sentence.
- Remaining Word Function – This assignment sends a sentence as a parameter to a function the student writes, and the student returns the remaining words after the first word of the sentence.
- Labs to practice programming lists:
- Max in List Function – This assignment sends a list of integers as a parameter to a function the student writes, and the function should return the largest integer in the list.
- Max Absolute in List Function – This assignment sends a list of integers as a parameter to a function the student writes, and the function should return the largest absolute value of each integer in the list.
- Average in List Function – This assignment sends a list of integers as a parameter to a function the student writes, and the function should return the average of all the integers in the list.

4.1 Derivates of Expression Labs

We modified the above labs that allow students to practice programming expressions so that each student received a derivative. Each of these labs initially provided the student with a formula or included an inherent method. So, for example, the Metric BMI lab provided students with the formula to calculate BMI by taking the weight in kilograms and dividing by the height in meters squared. The currency-based labs used an inherent method for converting the value of each coin. For example, a nickel is worth five pennies. All of these labs were modified by adding fake names for calculations and currencies and applying random constants and exponents. For example, one student would calculate the BMIA by using the formula of three times weight divided by two times height raised to the fourth power.

4.2 Derivates of Advanced Labs

After tackling the expression labs, we looked at the advanced programming labs listed above. None of these labs lend themselves to derivates in the problem statement. So we focused on ways to randomize the values in the unit tests to ensure integrity that the student was writing code that did not hard code output based on the inputs they saw. Each time a student submits their work, the test uses different input values that are randomly generated.

5 IT Course Auto-Graders

5.1 Helpdesk Course Auto-graders

In a helpdesk course, students learn technical problem-solving skills so they can solve end-user IT problems. We developed deployed through Docker sessions with problems and recipe-type instructions for the students to perform to solve the technical issues. We utilize the Linux Bash history file to auto-grade the student's work to ensure they executed all the commands in the recipe.

5.2 Networking Admin Course Auto-graders

Similar to the helpdesk course, the networking admin course teaches the student the core competency around network tools. We developed labs deployed through Docker sessions and provided recipe-type labs for the students to build familiarity with the networking tools. We utilize the Linux Bash history file to auto-grade the students' work to ensure they executed all the commands in the recipe.

6 Computer Science Course Auto-graders

6.1 Operating System Auto-graders

In an operating system course, students learn how operating systems manage limited hardware resources so that many application programs can run simultaneously. We developed auto-graders that allowed the students to explore the data structures and algorithms used to manage physical memory, virtual memory, hard disks, and the central processing unit (CPU).

6.2 Networking Programming Course Auto-graders

In a networking course, students learn about the Open Systems Interconnection (OSI) model and Transmission Control Protocol/Internet Protocol (TCP/IP) layers. The students write programs in Python that utilize TCP/IP services that talk to a cloud application.

7 Cybersecurity Science Course Auto-Graders

7.1 Information Security Auto-graders

In an information security course, students learn about threat modeling, security policy models, access control policies, and reference monitors. We developed a set of auto-graded reference monitor labs. In each lab, the student implements a reference monitor that implements different access control policies and security policies.

7.2 Secure Programming Auto-graders

In a secure programming course, students learn how to develop code free of vulnerabilities. The perspective in a secure programming course comes from the concept that the code is a white box. The students have full visibility of the source code as they perform labs to secure the code. We developed labs where students are provided code with vulnerabilities. Gradescope auto-graders are provided that exploit the vulnerabilities. Students need to improve the code and submit a version without the original vulnerability in their code to receive credit.

7.3 Penetration Testing Auto-graders

In a penetration course, students think about security from a different perspective. The perspective in a penetration testing course comes from the concept that the code is a black box. The students do not have visibility into the source code they are trying to penetrate in the labs. We developed labs where students are provided a signature for a code library with vulnerabilities. Gradescope auto-graders are equipped to execute the students' code and determine if they found a weakness.

8 Empirical Data

In this section, we examine the data we gathered from three sections of a database course. There were three questions we wanted to answer about our use of auto-graders in the cybersecurity curriculum:

- Do the auto-graders help students progress quicker through a lab?
- Do the auto-graders help increase participation at the undergraduate level in labs?
- Do the derivative auto-graders help students by facilitating peer discussion?

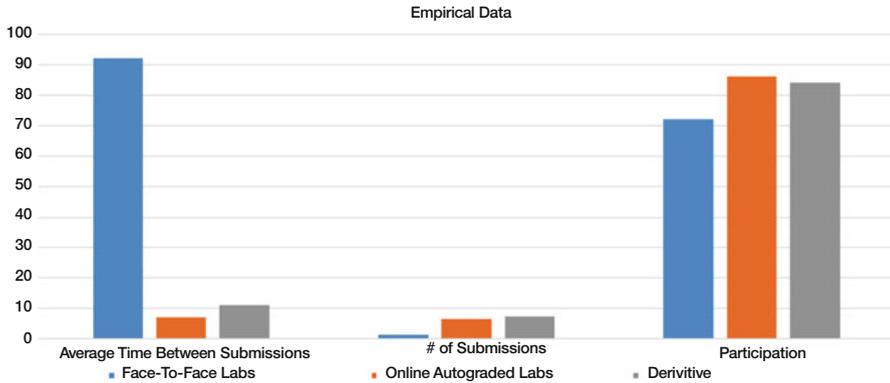


Fig. 2 Empirical data

We choose the database course because every lab had a derivative version so that each student was working on a unique problem in the lab. The original face-to-face section used manual graded lab submissions without derivatives, one online section used auto-graded nonderivative labs, and one section used the derivative labs. The students in the face-to-face section had a reverse classroom where they worked individually on labs during class time, and the instructor would answer questions as they ran into problems. In the section with the derivative labs, a discussion forum was provided for students to help each other with the lab.

Figure 2 shows a summary of the data we used to answer the questions. The average time between submissions was reduced significantly for the two sections that utilized auto-graders. The number of submissions was increased for the two sections that used auto-graders. Lastly, the participation rate was raised for the two sections that used auto-graders.

The answer to the three research statements was a strong yes to the first two and a weaker yes to the third question. The auto-graders helped online student progress quicker through the lab by shortening the time between submissions. In our small study, the auto-graders helped increase participation at the undergraduate level in both versions of the labs. Lastly, we believe the derivative auto-graders helped students by facilitating peer discussion. The participation rate was a little lower for the derivative version of the labs. Still, we felt it was close enough to the non-derivate lab to show progress in learning since students were performing unique work, and the increased student communication help to facilitate that progress.

9 Conclusions and Future Work

Based on our research, we demonstrate that the use of our e-learning auto-graded assignments improves participation in the cybersecurity course labs. We also show

that the use of our technique of creating derivatives of the lab for each student can lead to increased communication between students and therefore increased learning. Our future work will continue to develop labs in the advanced courses that not only randomize the unit test data but also provide for derivate problems per student. We will also gather more empirical evidence in the future to show how the auto-graders improve the learning experience for online e-learners.

References

1. S. Morgan, Cybersecurity talent crunch to create 3.5 million unfilled jobs globally by 2021, *Cybercrime Magazine*, 24 Oct 2019. [Online]. Available: <https://cybersecurityventures.com/jobs/>. Accessed 8 Apr 2020
2. NYU Tandon, NYU cyber fellows, 2020. [Online]. Available: <https://engineering.nyu.edu/academics/programs/cybersecurity-ms-online/nyu-cyber-fellows>. Accessed 8 Apr 2020
3. Georgia Institute of Technology, Online master of science in cybersecurity, 2019. [Online]. Available: https://info.pe.gatech.edu/oms-cybersecurity/?utm_source=cpc-google&utm_medium=paid&utm_campaign=omsc-search-converge-top5&gclid=CjwKCAjw7LX0BRBiEiwA__gNw2xT3-grxlfyfYGdGDGIpZZSkf6_blttgoipp830ue3le5MByUu0hoCGtoQAvD_BwE. Accessed 8 Apr 2020
4. Fisher College, Find the world at Fisher, 2020. [Online]. Available: www.fisher.edu. Accessed 8 Apr 2020
5. J.D. Ullman, Gradiance online accelerated learning, in *Proceedings of the 28th Australasian Conference on Computer Science*, (Newcastle, Australia, 2005)

The Effect of Matching Learning Material to Learners' Dyslexia Type on Reading Performance



Hadeel Al-Dawsari  and Robert Hendley 

1 Introduction

Learning disability is a general expression that covers disorders related to difficulties in various skills and senses. These difficulties may impact multiple abilities, such as reading, writing, listening, and speaking. Individuals, typically, suffer from such disorders due to a developmental issue. It is common for other factors such as sensory impairment and behavioral or other cognitive problems to co-occur with dyslexia [1].

Reading is one of the most important basic skills [2]. It can be considered as the gateway to learning other concepts. Most readers quickly learn to understand written text automatically and without any conscious effort. However, a percentage of readers face difficulties and tiredness when reading. This can lead them to be excluded socially and educationally, especially in classrooms [2]. Such readers are known as dyslexics. Dyslexia was first identified more than 100 years ago by Berlin [3].

Dyslexia is defined as a specific learning disability, widely believed to arise from a neurobiological issue. Dyslexics suffer from a disorder in the phonological component of language processing which leads to the following: (1) inaccurate and/or slow word recognition, (2) misspellings, (3) poor decoding ability [2], (4) word repetition or addition, and (5) character deletion and transposition [4].

H. Al-Dawsari

Department of Computer Sciences, Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia

H. Al-Dawsari (✉) · R. Hendley

School of Computer Science, University of Birmingham, Birmingham, UK

e-mail: hmalateeq@pnu.edu.sa; R.J.Hendley@cs.bham.ac.uk

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_60

837

Myklebust was the first person to suggest classifying developmental dyslexia into different types [5]. These early classifications tended to classify dyslexia based upon symptoms. This helps in understanding the difficulties and thus in providing more appropriate support [5].

On the other hand, [5] suggested classifying dyslexia based on the dual-route model for single-word reading and predicting dyslexics' symptoms based on problems with components of this model. This model has proved the most effective and widely used for predicting the various types of dyslexia [6]. Therefore, this classification of dyslexia is adopted in this research.

The potential of online learning has increased due to the emergence of new technologies. Online learning can be defined as accessing learning materials via the Internet in order to interact with the instructor, the learning content, and other learners. Among the benefits of online learning are that the learner can access it independently of time or location [7].

However, this way of learning will not, necessarily, provide learning material that suits an individual's needs. Each dyslexic is different and thus should be offered learning material matched to their individual needs. Adapting online systems has the potential to achieve this and to improve the quality of learning and the user's experience. Generally, adaptation means a procedure for customizing something to the users' needs [3]. In learning, it is often described as organizing the learning to accommodate learners' differences [8] or to change its behavior based on learners' needs [9].

Dyslexics are affected by the language they are learning – the differences in language structure and orthography have a large effect on the difficulties that learners face. There has been little research into supporting dyslexic students in Arabic. This research targets teaching reading skills to young, native Arabic-speaking dyslexics.

The Arabic language is the sixth most spoken language in the world.¹ Over 200 million individuals speak Arabic as their first language. It is also used as a second language by millions of Muslims [10]. The cursive nature of the script, the use of diacritics, having multiple forms for a single letter (depending on the position within a word), and non-vowelized text are some of the particular problems when reading Arabic [11].

This research is concerned with the evaluation of the effects of adaptation based on vowel dyslexia (VD). The TrainDys system adapts learning material to learners' needs. The evaluation is in terms of a controlled experiment to investigate whether matching the learning material to the type of VD improves learners' learning and increases their satisfaction. VD and short vowel dyslexia (SVD) were chosen because of their frequency [6].

This paper is organized as follows: the related work is presented in the next section, followed by a description of the experimental design, and finally, the

¹http://www.ethnologue.com/ethno_docs/distribution.asp?by=size. Accessed on June 2020.

experiment's results are presented and discussed along with recommendations for future work.

2 Related Work

Few research studies have investigated the use of adaptive online education systems for dyslexia in the Arabic language. Some have developed applications, while others proposed frameworks and guidelines. In terms of applications, most studies use game-based techniques. A variety of evaluation methods have been used. Ouherrou et.al [12] developed a standalone educational game to assess dyslexics' skills in Arabic. The application was evaluated by specialists using heuristic evaluation, and a questionnaire was used to get feedback from dyslexic children and teachers. They found that the educational game was useful and that the learning process might benefit.

Another study by Al-Ghurair and Alnaqi [13] aimed to enhance dyslexics' short-term memory through a game-based application structured around a story. Its usability was assessed through observation of children's interaction and of their use of the system. The children's opinions were taken after using the system. They report that the children were satisfied with the interface's theme and that the children enjoyed the application and were engaged.

There has been some work on tools related to dyslexia in Arabic. Aldabaybah and Jusoh [14] proposed a set of usability features to improve assistive technologies for Arabic dyslexics. They applied these in a prototype which was then evaluated by a special education expert. They suggest that the added features increased the students' perception of usability and enhanced their academic performance. Benmarrakchia [15] proposed a set of design guidelines based on dyslexics' spelling errors. These guidelines covered four areas: visual ability, phonological processing, orthographical similarity, and cognitive processing. However, the evaluation of these guidelines is left to later studies. AlRowais [16] developed a framework for the evaluation of training tools for Arabic dyslexics. This uses experts, interviews, and questionnaires. The evaluations were mostly positive, but they did identify unnecessary elements, gaps, and necessary refinements.

Overall, there is limited research into technological support for dyslexic education in Arabic. There are many gaps, and most of the work treats dyslexia as a single class. However, it is clear that there are many different classes of dyslexia and that these each require different adaptations. Finally, in terms of evaluation, most research either has a very limited evaluation or none, at all. Qualitative approaches (such as interviews and observations) are most common with very few quantitative and controlled studies and little that assesses the learning gain or the effect on students' satisfaction.

3 Method

In order to investigate the effects of matching learning material to learners' dyslexia type, we developed a training system (TrainDys) and used this to run a controlled experiment to assess learning gain and learner satisfaction.

3.1 Setup

The TrainDys system was designed and developed. There are eight exercises each with ten levels giving a total of 80 words. Most dyslexics have a visuospatial/kineshetic style [17], so a multisensory approach was used [15]. For each exercise, a word is spoken by the system, and its image is displayed along with three choices. If they choose the correct word, positive verbal feedback is given [15], and they gain a point. Otherwise, they are asked to try again [18]. The learner needs to achieve 80% [18] to unlock the next level.

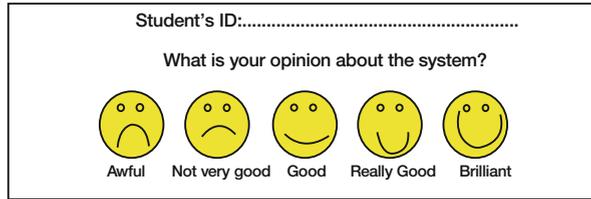
The material was chosen from the primary school curriculum to target vowel dyslexia. It uses a combination of short vowels (fat-ha, dammah, and kasra) and long vowels (a, i, u), (أ، إ، ؤ) progressing from simple (words of three letters and only fat-ha short vowel and alef long vowel) to advanced (five or six letters and a mix of the three short vowels and the three long vowels).

The cognitive theory of multimedia learning was used. The underlying theory was drawn from dual-coding theory, cognitive load theory, and constructivist theory [19]. Mayer and Moreno suggest five instructional design principles to achieve this: multimedia aids, contiguity aids, coherence aids, modality aids, and redundancy aids [19]. Each word is spoken and an image displayed (multimedia and modality aid), simultaneously (contiguity aid). The text was used to present the learning material (redundancy aid). No extraneous words or sounds unrelated to the learning material were presented (coherence aid). The TrainDys interface was designed following the guidelines for web design accessibility for Arabic content [20]. The experiment used these instruments:

- Diagnostic test: for diagnosis of dyslexia type.
- Consent form: parental approval since participants are under 18 years.
- Pre- and posttests: the first test to measure the knowledge level of the learner before using the system. The posttest was conducted after the course and used to calculate learning gain. Each test included ten words targeting VD (ten words of different lengths containing long and short vowels). For the posttest, a mixture of seen and unseen words was used.
- Satisfaction questionnaire: Because the subjects were young children, the smileyometer [21] was used (see Fig. 1).

Twenty learners were recruited. Due to the coronavirus pandemic, schools were closed partway through, and only 13 students were able to complete the

Fig. 1 Smileyometer [21]



experiment in school. Zoom meetings were used to complete the study remotely with 3 additional students, giving a total of 16. The remaining four learners did not complete the study for the following reasons:

- One learner’s parents refused remote participation.
- One learner refused, thinking they were on holiday.
- One learner had a poor Internet connection.
- One learner did not have a suitable device to conduct the experiment.

3.2 Procedure

The experiment was conducted in eight experimental sessions, each of 35 minutes. The study took place in Riyadh, Saudi Arabia. The experiment took place in person, in quiet rooms, in school. The three exceptional cases started as normal and were completed remotely using Zoom. The learner was first introduced to the study and completed the diagnostic test and pretest. They then used the TrainDys system. After finishing the study, they completed the posttest and satisfaction questionnaire.

4 Results and Discussion

The experiment was conducted with 16 female dyslexic learners; 8 were assigned to the SVD group (mismatched) and the other 8 to the VD group (matched). The learners were homogeneous in terms of knowledge level, age, and grade. They were in the fourth, fifth, and sixth grades in primary school. The mean age was approximately 10 (SD = 1.41) and ranged from 9 to 14. They were encouraged to take part in order to improve their reading.

4.1 Learning Gain

This shows that the learning gain of both the matched and the mismatched group was positive (posttest > pretest). This indicates that the reading of the matched group was

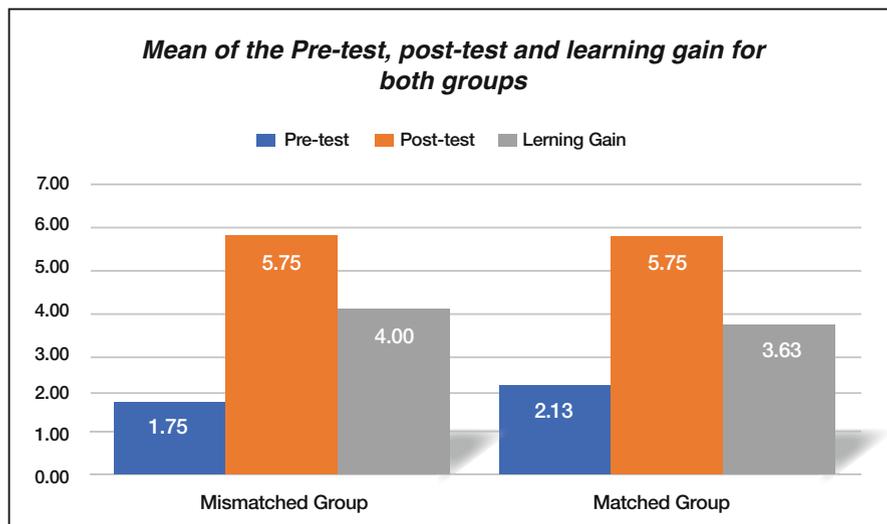


Fig. 2 Mean of the pretest, posttest, and learning gain for both groups

Table 1 Mean and standard deviation of the pretest, posttest, and learning gain

Group type	N	Pretest		Posttest		Learning gain	
		Mean	SD	Mean	SD	Mean	SD
Mismatched (SVD)	8	1.75	2.05	5.75	2.25	4.00	2.00
Matched (VD)	8	2.13	1.73	5.75	1.75	3.63	1.41

improved. This was also true with the mismatched group. The learning gain of both groups was, effectively, the same, and the posttest results were identical (Fig. 2). This result was surprising. It might be due to:

- The learning material not being new to either group: the main goal of the system was the reinforcement of a skill that had been taught earlier.
- The words used contained both long and short vowels. This will also benefit the mismatched group who have problems only with short vowels.
- The small number of participants.

The significance of the learning gain between the two groups was tested. As the data of the mismatched group was not normally distributed, as assessed by the Shapiro-Wilk test ($p = 0.034 < 0.05$), the Mann-Whitney U test was used as an alternative to an independent sample t-test. The difference in median learning gain for the matched group (3.50) and mismatched group (3.00) was not statistically significant: $U = 33.5$, $z = 0.163$, and $p = 0.878$.

However, these findings are true only for this sample of students and cannot be generalized due to the small sample size (Table 1).

Table 2 Mean, median, and standard deviation of the learner satisfaction

Group type	N	Learner satisfaction		
		Mean	Median	SD
Mismatched group (SVD)	8	4.50	5.00	1.41
Matched group (VD)	8	4.75	5.00	0.71

4.2 Learner Satisfaction

Table 2 shows the mean, median, and standard deviation of the learners' satisfaction for each group. The mean for both groups was almost the same, and their median was identical. The scores indicate that learners in both groups were very satisfied with the system, and so, it is not possible to detect any meaningful effect. This could be due to the interactive feature of the system and the guidelines that were followed during system design [20]. Again, these findings are true only for this sample of students and cannot be generalized due to the small number of subjects.

5 Lessons Learnt

During this experiment, several lessons were learned. It is very hard to recruit large numbers of students with an appropriate profile. We restricted participants to have a specific form of dyslexia, to be in one geographical area (for practical reasons), and to have similar ages and reading performance. The problems that arose through the coronavirus pandemic did disrupt the study. However, this also highlighted that it may be possible to conduct future studies remotely – and therefore to recruit a much larger number of participants across a wider geographical area.

In terms of the TrainDys system, there were several issues. The learning material did not discriminate sufficiently between the needs of the different students. There was also too steep a progression in the difficulty of the material. Each of these factors will be addressed in future work.

The instruments used also had some weaknesses and need to be refined. A particular problem was with the assessment of the students' satisfaction. The scores given by the students were extremely high. While this is reassuring, in terms of the quality of the learning material, it does mean that it is impossible to discriminate between the conditions. Again, this will need to be reassessed and refined.

6 Conclusion and Future Work

Dyslexia is a universal reading difficulty. It can be found everywhere and independently of language. However, just as everybody is different, dyslexics are too. They suffer from different reading problems. For instance, some of them may not

understand what is written, while others may omit, transpose, or alter letters while reading.

The aim of this research is to overcome these problems by providing dyslexic individuals with appropriate learning material that matches their needs. It is expected that this will improve their learning and their satisfaction. This, in turn, is expected to improve their engagement, which should have further benefits.

We designed a controlled experiment to test this. The students' dyslexic type was diagnosed, and their prior reading performance was assessed. They were then assigned to one of two conditions. Learning was delivered through an online system, and their reading performance and satisfaction were assessed after completing the course. The results showed an improvement in learners' reading performance in both conditions and very high levels of user satisfaction.

There has been little research into using adaptive learning for dyslexia in Arabic. This work seeks to explore the feasibility and benefits of this approach. The results of this preliminary study do not clearly demonstrate any benefit. Our further work will explore making the adaptation more useful and also expanding the number of participants.

Acknowledgments The authors thank Princess Nourah Bint Abdulrahman University, Riyadh, Saudi Arabia, and the University of Birmingham, United Kingdom, for their support.

References

1. D. Hammill et al., A new definition of learning disabilities. *J. Learn. Disabil.* **11**(3), 217–223 (1988)
2. M. Mastropavlou, V. Zakopoulou, Integrated intelligent LEARNing environment for reading and writing D3.2 – Learning strategies specification report. *Work* **4** (2013)
3. P. Brusilovsky, Adaptive hypermedia for education and training. *Adapt. Technol. Train. Educ.* **46**, 46–68 (2012)
4. M. Tafti, M. Hameedy, N. Baghal, Dyslexia, a deficit or a difference: Comparing the creativity and memory skills of dyslexic and nondyslexic students in Iran. *Soc. Behav. Personal. Int. J.* **37**(8), 1009–1016 (2009)
5. N. Friedmann, M. Coltheart, Types of developmental dyslexia, in *Handbook of Communication Disorders: Theoretical, Empirical, and Applied Linguistics Perspectives*, (MDPI, 2016), pp. 1–37. <https://www.mdpi.com/2076-3425/10/11/896>
6. N. Friedmann, M. Haddad-Hanna, Types of developmental dyslexia in Arabic, in *Handbook of Arabic Literacy*, (Springer, 2014), pp. 119–151
7. M. Ally, Foundations of educational theory for online learning. *Theory Pract. Online Learn.* **2**, 15–44 (2004)
8. G. Magoulas, Y. Papanikolaou, M. Grigoriadou, Adaptive web-based learning: Accommodating individual differences through system's adaptation. *Br. J. Educ. Technol.* **34**(4), 511–527 (2003)
9. K. Feigh, M. Dorneich, C. Hayes, Toward a characterization of adaptive systems: A framework for researchers and system designers. *Hum. Factors* **54**(6), 1008–1024 (2012)
10. A. Mahfoudhi, J. Everatt, G. Elbeheri, Introduction to the special issue on literacy in Arabic. *Read. Writ.* **24**(9), 1011–1018 (2011)

11. G. Elbeheri, Dyslexia in Egypt, in *The International Book of Dyslexia: A Guide to Practice and Resources* (John Wiley & Sons, 2005), pp. 79–85
12. N. Ouherrou et al., A heuristic evaluation of an educational game for children with dyslexia, in *2018 IEEE 5th International Congress on Information Science and Technology (CiSt)*, (IEEE, 2018), pp. 386–390
13. N. Al-ghurair, G. Alnaqi, Adaptive Arabic application for enhancing short-term memory of dyslexic children. *J. Eng. Res.* **7**(1) (2019)
14. B. Aldabaybah, S. Jusoh, Usability features for Arabic assistive technology for dyslexia, in *2018 9th IEEE Control and System Graduate Research Colloquium (ICSGRC)*, (IEEE, 2018), pp. 223–238
15. F. Benmarrakchi, J. El Kafi, A. Elhore, Communication Technology for users with specific learning communication Technology for users with specific learning disabilities, in *FNC/MobiSPC*, (Elsevier, 2017), pp. 258–265
16. F. AlRowais, M. Wald, G. Wills, Developing a new framework for evaluating Arabic dyslexia training tools, in *International Conference on Computers for Handicapped Persons*, (Springer, Cham, 2014), pp. 565–568
17. S. Exley, The effectiveness of teaching strategies for students with dyslexia based on their preferred learning styles. *Br. J. Spec. Educ.* **30**(4), 213–220 (2003)
18. H. Lyytinen et al., In search of a science-based application: A learning tool for reading acquisition. *Scand. J. Psychol.* **50**(6), 668–675 (2009)
19. R. Mayer, R. Moreno, Aids to computer-based multimedia learning. *Learn. Instr.* **12**(1), 107–119 (2002)
20. A. Al-Wabil, P. Zaphiris, S. Wilson, Web design for dyslexics: Accessibility of Arabic content, in *International Conference on Computers for Handicapped Persons*, (Springer, Berlin, Heidelberg, 2006), pp. 817–822
21. J. Read, Validating the fun toolkit: An instrument for measuring children's opinions of technology. *Cogn. Tech. Work* **10**(2), 119–128 (2008)

Individualized Educational System Supporting Object-Oriented Programming



F. Fischman, H. Lersch, M. Winterhagen, B. Wallenborn, M. Fuchs, M. Then,
and M. Hemmje

1 Introduction, Motivation, Problem Areas, and Research Questions

With regard to software developers who have made a huge impact in every industry field, one would require more qualitative developers [1], and therefore, a new learning technique is needed to study OOP [2] material more effectively and efficiently. This learning technique that is currently being developed will be known as the adaptive competence-based educational system (ACEDuSys), which will entail adaptive learning [3] and competence-based education [4] for the purpose of learning and understanding the learning material. Also, the learning material will explain the descendants of OOP languages in addition to object-oriented language functionality. OOP is being incorporated worldwide across educational institutions as an essential element of modern education [5], and the majority of students experience great difficulty studying and understanding the logic and syntax [6]. Learning to program is complicated, and the dropout rate is high; [6] therefore, a new effective result is necessary to find better teaching methods. To improve the effectiveness of the traditional way of teaching OOP, adaptive learning and competence-based education needs to replace the traditional way of learning, which will lead to innovative teaching methods and management infrastructures.

F. Fischman (✉)
New York Institute of Technology, New York, NY, USA
e-mail: ffischma@nyit.edu

H. Lersch · M. Winterhagen · B. Wallenborn · M. Then · M. Hemmje
FernUniversität in Hagen, Hagen, Germany

M. Fuchs
Wilhelm Büchner Hochschule, Darmstadt, Germany

The motivation of this paper is to address the educational system, which will discuss reasons for failure to comprehend OOP concepts, pedagogical methods increasing the success rate of understanding OOP philosophy, and ways on how to speed up the process of learning OOP concepts. Furthermore, another incentive is to make available an innovative LMS [5] that supports personalized learning paths, which is known as adaptive learning. Based on the above introduction and motivation, different problems areas will be elaborated where the *Problem Area 1 (PA1)* is that learning OOP is ineffective and inefficient. *Problem Area 2 (PA2)* is that currently teaching OOP is not individualized and majority of the traditional learning or one-size-fits-all models are not using adaptive learning and competence-based education. From PA1 and PA2, the Research Question 1 (RQ1) and Research Question 2 (RQ2) are to follow: (RQ1) *How can OOP teaching be made more effective and efficient?* (RQ2) *How can OOP teaching be adaptive and individualized?* RQ1 will lead to the first challenge, and that is to clear up whether there exists an educational system that can make teaching OOP more effective and efficient. Furthermore, RQ2 will lead to the second challenge, and that is to clear up whether there exists an educational system that uses an individualized approach to teach OOP.

2 Methodology, Goals, Approach, and Outline

In this section, the research questions from Sect. 1 will be used to apply our research method and how the research methodology can result in achieving the research goals. Our methodological meta-model is based on Nunamaker [7], which consists of four types of research and development activities such as observation, theory building, system development, and experimentation. We will apply each of the research activities to our research questions as follows: The RQ1 will lead to the overall objective of the Research Goal 1.1 (RG1.1), and that is to perform an observation study as to why the OOP is ineffective and inefficient. Furthermore, RQ1 will lead to the overall objective of the RG1.2, and that is to perform an observation literature study on what type of learning methods can make OOP more effective and efficient. The RQ1 will lead to the overall objective of the RG1.3, and that is building a theory on how to make teaching OOP more effective and efficient. The RQ1 will lead to the overall objective of the RG1.4, and that is to develop a prototype which will entail details of the system for a proof of concept. The RQ1 will lead to the overall objective of RG1.5, and that is to add an evaluation for improving the prototype on ineffectiveness and inefficiency. The RQ2 will lead to the overall objective of the Research Goal 2.1 (RG2.1), and that is to perform an observation study as to why the OOP teaching is not adaptive and individualized. Also, the RQ2 will lead to the overall objective of the RG2.2, and that is to perform an observation literature study on how to make teaching OOP adaptive and individualized. The RQ2 will lead to the overall objective of the RG2.3, and that is to build a theory using models as to how to make teaching OOP adaptive and

individualized. The RQ2 will lead to the overall objective of RG2.4, and that is to develop a prototype which will entail details of the system proof of concepts. The RQ2 will lead to the overall objective of RG2.5, and that is to add an evaluation for improving the prototype on adaption and individualized teaching of OOP. The outline of this paper will follow the approach by first introducing the observation results in Sect. 3, which deals with RQs from the start and explains which approach is most suitable for achieving the RGs so it can be structured accordingly. Section 4 will present the conceptual model with proof of concept, and finally, Sects. 5 and 6 will extant the prototype implementation and evaluation of the ACEDuSys.

3 State of the Art in Science and Technology

According to the European Qualification Framework (EQF) [8], having a common framework across universities for the purpose of students transitioning between various countries for the purpose of skill and knowledge comparability, one requires the implementation of EQF [8] more precisely the e-Competence Framework (e-CF) [8], which was established, e.g., supporting the needs for information and communication technology (ICT) industry in Europe. The identification, description, and representation was released as a current version of the e-CF. The description of competencies took about ten years to define and was released as the common European framework. Competence-based learning considers competencies for teaching toward a specific learning goal to facilitate the learning process more effectively and efficiently. The so-called competence-based learning from the European funded research project TENCompetence provides the Personal Competence Domain Model (PCDM) [8], which enables modeling of various scenarios with standardized competencies. One particular scenario of competence-based learning would be, e.g., that a student takes a course in Basic Java Programming (BJP) where the topic of objects is covered. After the completion of the course, the student might require transitioning into another OOP language like Basic Python Programming (BPP) course meaning the knowledge of objects can be transitioned from the BJP course into the BPP course and only the syntax will vary. Furthermore, after the completion is established, the OOP learning becomes individualized and adaptive to the preexisting knowledge of the learner. Competence-based learning aids the learning process toward a learning goal more effectively and efficiently, and this knowledge that was obtained can be used toward establishing the so-called personalized learning path [8]. The TENCompetence project designed the entire learning process based on competencies where the PCDM supplies machine-readable taxonomies mapping and was enhanced further by Qualification-Based Learning Model (QBLM) [8], which supports the use of taxonomies. PCDM originated from the TEC Competence project and afterward became a more general model and was called QBLM, and this model is able to link qualifications to EQF. The QBLM enhanced the PCDM, which provided a class competence to map qualifications but without displaying qualification instance and scope, which was one of

the requirements of e-CF [8]. Following the concept, the entire representation based on QBLM can be related to the so-called KM-EP [10] and its underlying learning management system (LMS) [8] Moodle [8] by providing a brief summary of what actually was accomplished in [8]. According to [9] in the KM-EP, one operates with competences that were based on [10], and in the LMS Moodle, one operates with qualifications with the help of the QBL Plugin for Moodle (QBLM4Moodle) [10]. In order for the KM-EP and the Moodle to be compatible with each other, a more specific data point mapping was developed so competencies and learning activities could be imported from the KM-EP into the LMS Moodle, which was created by [8]. Perhaps, a future topic that could be addressed in another research activity would be the development of an inverse mapping. Again, the idea behind [8] was to implement an extension of the KM-EP, the so-called Course Authoring Tool (CAT) for the purpose of adding competencies with learning activities via an interface in order to enable the export possibility of competencies into the LMS Moodle. Further, the related research conducted on the basis of QBLM enables the creation of learning processes by applying machine-readable qualifications. To enhance the development further, a foundation requirement needed to be laid out for adaptive courses, automated recommendations, and services, which are called structured learning templates (STP) [10]. To make the basis more efficient, it is necessary to have competency information represented down to the smallest single learning activity, which will make available for various courses, internal and learning units [8]. The KM-EP is the overall web-based knowledge management system which operates with competences that were based on the TENCompetence project in one of its sub-system of the LMS Moodle.

With regard to [9], an application of normalization of competency information was enhanced into KM-EP where [10] developed a QBLM4Moodle plugin in Moodle for a similar reason; therefore, one can state that both applied different implementations to solve the same problem in different software. The CAT, which is part of KM-EP, is connected to Moodle, and therefore, courses can be explained and commented on or annotated using competency information. Furthermore, a course needs a specific competence coming from students as a prerequisite competence in order to begin to work on it, and once the students have passed a course, a specific competence was accomplished by them. There are two types of information available about competencies for a course like a precondition or requirement and a goal or desired result competency. The QBLM4Moodle plugin enables the course author in Moodle to assign competence requirements as a precondition or condition to occur before other things can happen and competence to obtain as a goal in Moodle courses. This paper introduces the solution on how this gap will be filled. It is based on the research and development activities described by [8]. The following related to the section theory building based on concept work will demonstrate how this gap will be solved conceptually.

4 Conceptual Modeling

In this section, the conceptual modeling will be presented based on the theory building involving a user-centered system design [16] and will be formalized using the Unified Modeling Language (UML) [11] that will entail use cases, classes, and sequence diagrams, and a brief explanation will follow on mapping for the purpose of providing a basic background on the details that were established in [8]. UML use case diagrams provide a general picture what is happening in the existing system or is planned to occur in the new system. The use case diagrams show use cases and actors and the association between them. Use cases represent sequences of actions, and actors represent the people or other systems that interact with the system being modeled [11]. For example, in Fig. 1, the human symbol is the user role name; in this case, it is course author, and the ellipses with text inside represent the use cases. The solid line between the course author and the use case represents the association between them. Furthermore, the include relationship is represented as an open arrow with a dashed line, which points toward the use case that is being modeled, and the word include is written in guillemets on the relationship arrow shown in Fig. 1. The <<include>> means that the use case edit activity is a use case by itself and is included into the use case assign precondition profile (CP). The same breakdown logic is applied to Figs. 2 and 3 [8, 11].

A course author can use the CAT to create new courses, which can be divided into sections and contain learning activities. Furthermore, a course author can assign a CP as a condition or a goal profile (GP) to a course. In addition, a CP can be assigned to a learning activity where the user is able to update the condition or GP for courses by applying the useProfileAction() method for transferring condition or GP information for courses. To update an activity, the updateActivity() method applies services to transfer competence or GP information for an activity into the LMS [8]. By applying these use cases, which demonstrates efficient communication between users and system, one can associate a condition or GP to an activity. A further

Fig. 1 Use case to add condition or goal profile to an activity

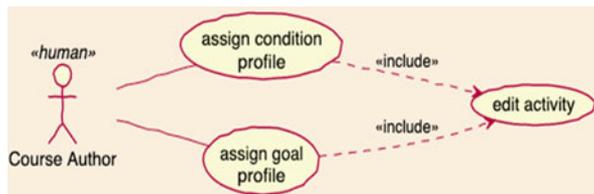


Fig. 2 Use case to transfer condition or goal profile of a course to an LMS

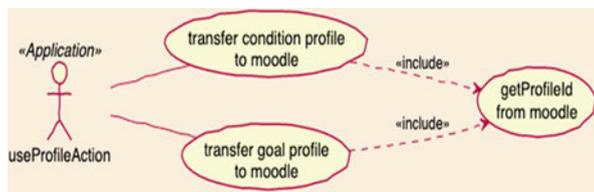


Fig. 3 Use case to transfer or goal profile of an activity to an LMS

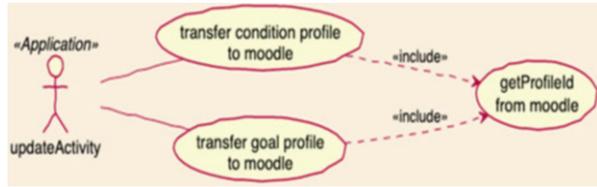
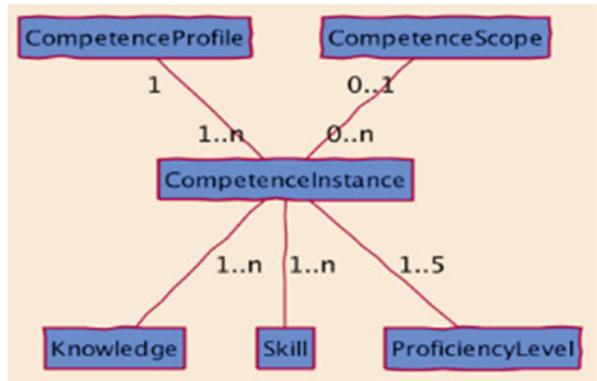
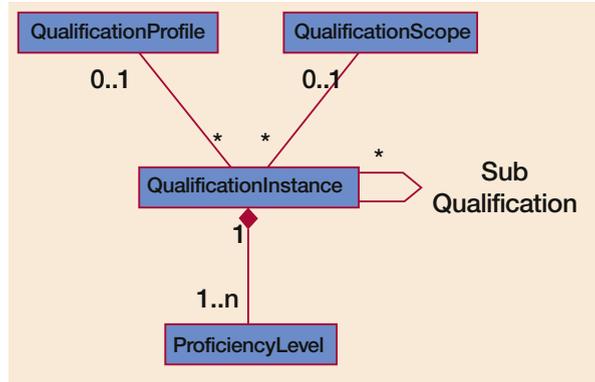


Fig. 4 Competence scope



enhancement will allow a condition profile and goal profile editing an activity. The course author should be able to select competence profiles during the editing of an existing activity, which is shown in Fig. 1 above [8]. When the updateActivity() method is invoked, the CP or GP should by default be transferred into the LMS and connected to the activity after verification existence of the profile. Furthermore, the updateActivity() method needs to be enhanced with regard to checking the existent profile in LMS to transfer the competency instances, which is shown in Fig. 3 [8]. To accomplish RG2, one requires knowledge on how to handle competence profiles, and this is accomplished via UML domain class diagrams. A domain class diagram shows the entities and the relationship between them [17]. By looking at Figs. 4 and 5, one can see the rectangles with text represent the entities and solid lines represent the links. The multiplicity of domain class diagrams can be explained such as 1 . . . * one or more, 0 . . . 1 zero or one, and 1 exactly one. Each multiplicity has a bound range [m . . . n] where m is the lower and n is the upper limit range. The solid line with a diamond means “has a” and is also known as aggregation [11]. Furthermore, a competence is an outcome, an instance is a single occurrence of something, and a profile is an outline of something or more specific it is information about competence instances. In Fig. 4, a CP is a precondition students have to meet to start a course, and by passing the course, students will gain new competences [8]. A competence instance (CI) is a particular competence with a specific skill, knowledge at a specific proficiency level (PL) [8]. A qualification or an accomplishment is used in Fig. 5 to explain qualification scope (QS) and qualification profile (QP). One example from Fig. 5 would be that the one or more proficiency level has a single qualification instance. Note that an instance is a subset of a scope and a set

Fig. 5 Qualification scope



of instances make up a profile. It is necessary to map the CompetenceScope to a QualificationScope for the purpose of showing competency profile as a qualification profile. A qualification scope in Fig. 5 is different from a competence scope in Fig. 4; it is a general object of type competence, knowledge, or skill. One can state that a composition of qualification scope contains a minimum of one proficiency level, which is shown in the domain class diagram in Fig. 5 [8].

To enhance the system design further, one requires the usage of UML sequence diagrams, which are used to demonstrate how objects interact with each other to complete a task by drawing a rectangle box to represent the role play of an object labeled by the name of the class for the object. Furthermore, vertical dotted lines or lifelines are used to represent an object as time passes by, and solid line arrows are used to represent messages that are sent from one object to another (sender to receiver) [11]. To return data, one can use the dotted line arrow (receiver to sender), and to activate the object, one can use small rectangles on the objects' lifeline that happens when an object sends or receives messages. It is possible to take the sequence as part of an alternative or offering a choice process meaning a sequence of actions will occur if a condition is true by putting the sequence in a box and label its alt for alternative in the top right corner. Furthermore, it is necessary to specify when this alternative will occur, meaning a sequence occurrence is true. If the sequence does not occur, then other sequences can occur, which is dragged underneath the previous sequence with the condition else; however, this sequence can contain a loop, which can be labeled as a box loop in the top right-hand corner [11]. This logic is applied to Figs. 11 and 12 in the Annex section. In [8], it demonstrates the requirement implementation to transfer profile information that is connected to a course and an activity. In order to transfer a condition profile or a goal profile, it requires that profile information of a course is updated, and the implementation transfer is shown in Fig. 11. Steps 1–3 in the qualification profile in Moodle will be determined by invoking the web service `get_profile_id` via KM-EP service `getProfileId`. Suppose a current profile exists, then in Fig. 11, steps 4–6 will demonstrate how it will be emptied, and else steps 7–9 in Fig. 11 will create a new profile. In the empty profile, steps 10–15 in Fig. 11 will show

how it will create a lot of qualification instances as competence instances. Figure 12 demonstrates the transfer process into Moodle, which occurs after associating a condition profile or goal profile to an activity [8]. During the update, steps 1–3 in Fig. 12, the service updateModule invokes the service update_module in the LMS in order to transfer the information activity into an LMS unit. Suppose a current profile exists then in Fig. 12, steps 4–6 occur by invoking a web service get_profile_id via the getProfileId. If for some reason the occurring profile is discovered, then steps 7–8 in Fig. 12 will be emptied, and the else steps 10–12 in Fig. 12 demonstrates how a new profile is established. The empty profile steps 13–18 in Fig. 12 will illustrate how it will create a lot of qualification instances as competence instances [8]. In order to understand the implementation, functionality to export linked competences from a course and its learning activities from the mapping is required. It is possible to relate QBLM mapping to the LMS database with the aids of extension points and the application programming interface (API), which are available in the LMS [8]. One of the tasks of QBLM4Moodle plugin is the development of the LMS extension without having to modify the core program code, because many extension points and APIs are provided [8, 10]. The QBLM4Moodle plugin enhances the LMS by splitting qualifications into qualification scopes and instances, which are the building blocks of QBL [8]. Based on the conceptual work in this section a prototype implementation will be presented in the next section.

5 Prototype Implementation

Differently based technologies were used to implement the proof of concept for the research goals from Sect. 2 such as Hypertext Preprocessor (PHP) [12] suited for web design at the back end, Symfony a web application framework [13], Doctrine PHP Object Relational Mapper (ORM) [14] which allows writing database queries object-oriented SQL called Doctrine Query Language (DQL), MySQL [15] a database management system (DBMS) for operating on a set of tables with data at the back-end, KM-EP which is a portal that offers knowledge management tools and web-based authoring tools for courses, and Moodle, an LMS used for pedagogical principles, distance learning, and e-learning.

Based on the use cases and domain class diagrams in Sect. 4, the accomplished research goals that is the association of a competence profile and goal profile to learning activities can be followed in [8] in details using the KM-EP, which is the implementation of RG1. In order to accomplish RG2 that is exporting competency information from the KM-EP into the LMS Moodle, one needs to be aware that KM-EP is able to export information about courses via web services into the LMS Moodle. The detailed steps are laid out in [8] on how it is done using KM-EP. The accomplished RG2 which deals with data point mapping is shown in the UML object diagrams in Figs. 6. and 9 [8].

Note that in KM-EP, one operates on competences where in the LMS Moodle, one operates on qualifications, and therefore, in Figs. 6 and 7, one can see that KM-

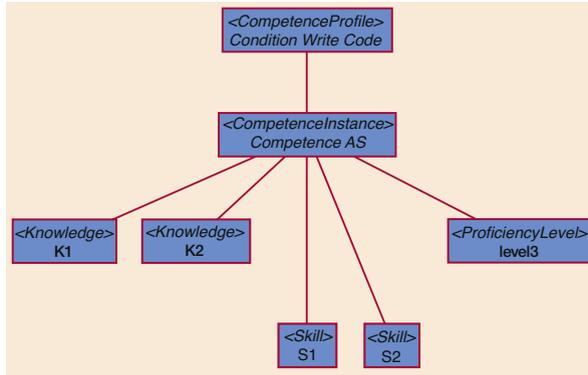


Fig. 6 Competence profile writing code in KM-EP data model

EP contains various objects like competence, knowledge, and skill where the LMS Moodle has only one object QS of type competence, knowledge, and skill as shown in Fig. 7. Furthermore, in Fig. 7, the types are added into the UML object diagram like so `<featureName>:<type>`, which are data types returned by an operation. One simple example would be that in Fig. 6, the `CompetenceInstance` `Competence A5` is equivalently transferred into LMS Moodle as `QualificationInstance A5` with a return-type competence shown in Fig. 7 [8]. The UML object diagrams can be converted to simple diagrams, which demonstrate the connection between Figs. 6 and 7. In Fig. 8, one can see that the CP user interface in KM-EP is structured like an upside-down tree where competences in KM-EP can be picked out first, which are expressed in the form of skill examples, knowledge examples, and PLs. In Fig. 9, in the qualification user interface in Moodle QBL plugin, one can see how the CPs are transitioned into Moodle where it contains only one QS. Figures 8 and 9 demonstrate the mapping from KM-EP to Moodle's QBL.

6 Evaluation of the System

To evaluate the system and its application, the cognitive walkthrough (CW) method [8] was used from the perspective of users, and the first step was to prepare the CW, which involved real examples on the usage of the system. Furthermore, the steps on how the real use case tasks were accomplished were displayed. The first CW was accomplished on a course from the Department of Mathematics and Computer Science at the University of Hagen [8]. This initial CW was used to demonstrate the functionality of the authoring tools, and in addition, it demonstrated how the material was created and linked to goals and preconditions in the form of competence instances using the CAT in KM-EP. In conclusion, it showed how to export it into the LMS Moodle [8]. The exact steps from the perspective of the

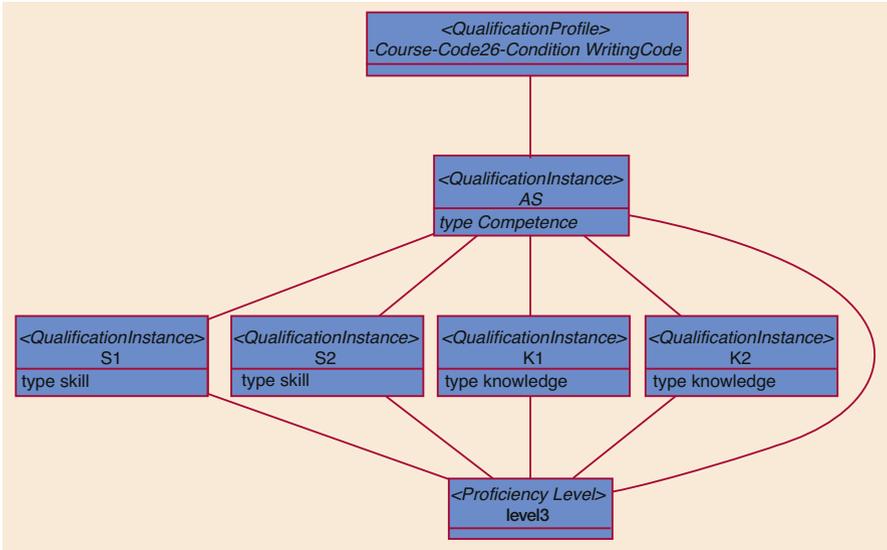


Fig. 7 Qualification profile writing code in QBLM4Moodle data model

Choose competences for profile Condition 1

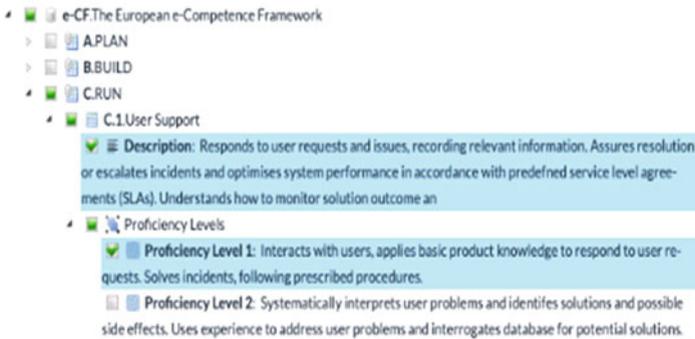


Fig. 8 Competence Profile User Interface in KM-EP

user can be followed in the completed work of [8], which have been validated by the accomplished RG1 and RG2 [8]. Based on the accomplishments of [8], a sub-goal will be used for evaluating the system further for a dynamic personalized learning path (DPLP) in order to make the learning of OOP more efficient and effective. The DPLP will assess students separately on each topic of a BJP course, and based on the outcomes, certain topics will be omitted, and certain topics will be required to be studied. Suppose a student is tested on knowledge of *objects*

Qualification Profile

[Back to list of profiles](#)

ID	Scope	Level
335	Skill-52 - S1 explain and communicate the design/development to the customer	2
336	B.1 - Application Development Interprets the application design to develop a suitable application in accordance with customer needs. Adapts existing solutions by e.g. porting an application to another operating system. Codes, debugs, tests and documents and communicates product deve	2
337	Skill-53 - S2 perform and evaluate test results against product specifications	2
338	Knowledge-57 - K1 appropriate software programs/modules	2
339	Knowledge-58 - K2 hardware components, tools and hardware architectures	2

Fig. 9 Qualification User Profile Interface in Moodle QBL Plugin

and the assessment demonstrates that the topic on *objects* can be omitted based on preexisting knowledge, the personalized course will be generated without the topic of *objects*. Furthermore, if this student decides to study another course like BPP, the knowledge of *objects* will be omitted from the course. Conceptually, this positive experimental validation will be tested and eventually applied to the LMS Moodle. By means of the second CW, the testing scenario is designed and shown in Fig. 10 [18].

7 Conclusion

The overall overview of this paper was to provide the reader with the work accomplished by [8] that has been implemented as an extension of the KM-EP CAT to associate competencies with learning activities and an interface to export competencies to the LMS Moodle with the aid of the QBLM4Moodle plugin for the purpose of enhancing the LMS Moodle further, which was described above in this position paper. The competences from KM-EP that can be exported into the LMS Moodle will be used for future work, which will deal with enhancing the LMS Moodle in terms of being able to generate a dynamic course based on individual student assessment using specific competences or outcomes to accomplish a DPLP by applying the CBE and AL to master the technical knowledge that is related to the OOP language. This DPLP will allow each student to transition across OOP languages very quickly, and that is an important asset or an advantage that the industry is constantly looking for specifically now as study and working conditions have changed due to unforeseen events in the world. Perhaps, this work will be discussed in the next position paper.

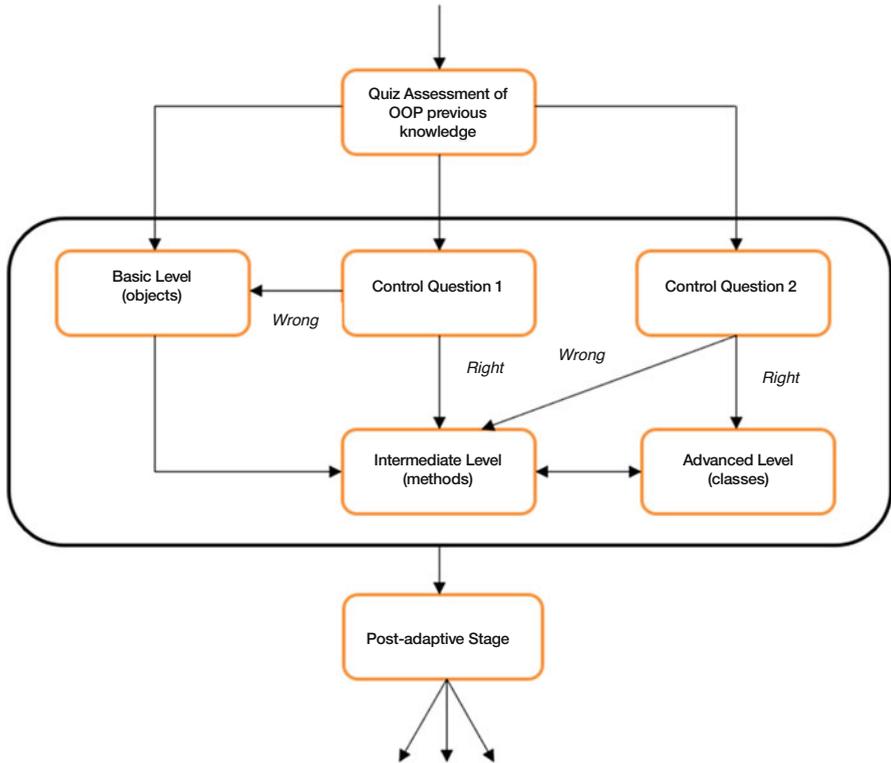


Fig. 10 Use case of dynamic personalized learning path

A.1 Annex

The UML sequence diagrams are shown below and are explained in detail in Sect. 4, Conceptual Modeling.

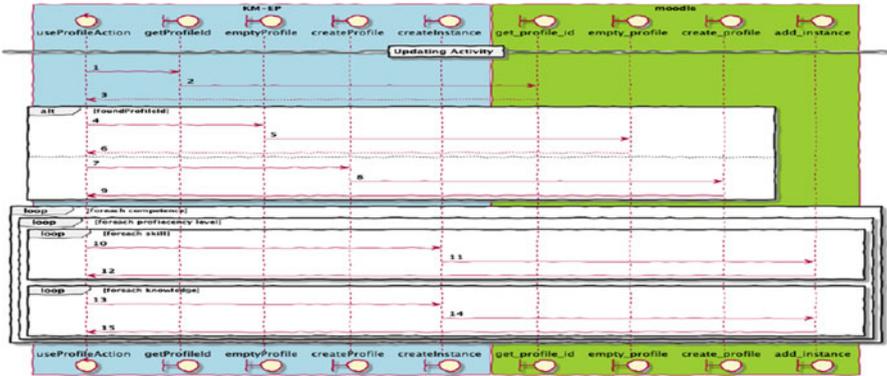


Fig. A.1 Transfer of competence information into a LMS for a course

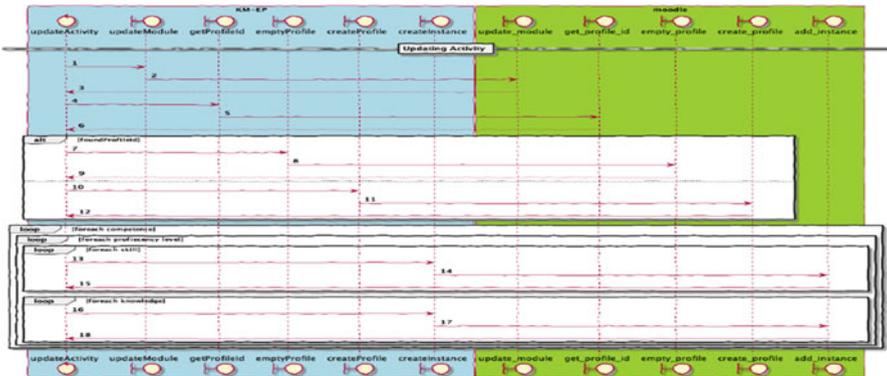


Fig. A.2 Transfer of competence information into a LMS for an activity

References

1. U.S. Bureau of Labor Statistics. <https://www.bls.gov/ooh/computer-and-information-technology/computer-programmers.htm#tab-6>, 6/23/20
2. OOP Concept for Beginners: What is Abstraction by Thorben Janssen. <https://stackify.com/oop-concept-abstraction/>, 6/23/2020
3. Knewton Adaptive Learning. <http://www.lmi.uib.edu/cursos/s21/REPOSITORIO/documents/knewton-adaptive-learning-whitepaper.pdf>, 6/23/2020
4. What is Competence Based Education. http://www.huffingtonpost.com/dr-robert-mendenhall/competency-based-learning-_b_1855374.html, 6/23/2020
5. Learning and Teaching Programming: A Review and Discussion by Swets & Zeitlinger. <http://www.science.smith.edu/classwiki/images/1/14/RobinsRev.pdf>, 6/23/2020
6. The Countries Introducing Coding into the Curriculum by Natali Vlatko. <https://jaxenter.com/the-countries-introducing-coding-into-the-curriculum-120815.html>, 6/23/2020
7. J.F. Nunamaker Jr., M. Chen, & T.D.M. Purdinn, System development in information systems research, J. Manag. Inf. Syst., Winter 1990–1991. <http://gkmc.utah.edu/7910F/papers/JMIS%20systems%20development%20in%20IS%20research.pdf>, 6/23/2020

8. Implement an extension of the KM-EP Course Authoring Tool (CAT) to associate competencies with learning activities and an interface to export competencies to the LMS Moodle
9. Development of an Innovative Authoring Environment for University Distance Learning, FernUniversität in Hagen, Germany, 2018
10. Supporting Qualifications Based Learning (QBL) in a Higher Education Institution's IT Infrastructure, FernUniversität in Hagen, Germany, 2020
11. S. Bennett, J. Skelton, K. Lunn, *Schaum's Outline UML* (McGraw-Hill, 2001)
12. PHP. <https://en.wikipedia.org/wiki/PHP>, 6/23/2020
13. Symfony. <https://en.wikipedia.org/wiki/Symfony>, 6/23/2020
14. Doctrine ORM. <https://www.doctrine-project.org/>, 6/23/2020
15. MySQL. <https://en.wikipedia.org/wiki/MySQL>, 6/23/2020
16. D.A. Norman, S.W. Draper, *User Centered Design New Perspective* (Lawrence Erlbaum Associates, 1986)
17. Silberschatz, Korth, Sudarshan, *Database System Concepts* (McGraw-Hill, 2006)
18. Daniel Burgos, Pablo Moreno Ger, and Baltasar Fernandez Manjon, *Building Adaptive Game Design-based Learning Resources: The Marriage of IMS Learning Design and <e-adventure>*, Sage Publications, 2007

Part VIII
e-Business, Enterprise Information
Systems, and e-Government

Emerging Interactions of ERP Systems, Big Data and Automotive Industry



Florie Bandara and Uchitha Jayawickrama

1 Introduction

Enterprise resource planning (ERP) systems play an increasingly important role in contemporary business technology management [1]. All the companies in the automotive industry share the common goal of performing streamlined and efficient operations in order to maximize profitability [2]. Therefore, many automotive companies adopt ERP systems to gain a competitive advantage in the demanding business environment [1, 3]. Big data is an extremely essential technology in the automotive industry as well as automobile manufacturing with the launch of innovative concepts day by day [4–6]. Despite the fact of emerging numerous technologies to increase the efficiency of the day-to-day activities, there are still critical issues coming up [3, 7]. In this paper, the rare connection of the two technologies ERP and big data with the automotive industry is created and identified under three categories of critical issues and has proposed a conceptual framework to control and minimize the issues.

In Sect. 2, the paper discusses the conducting of the systematic literature review (SLR), whilst Sect. 3 demonstrates the key findings of the SLR discussing the connection of big data and ERP systems with the automotive industry. Moreover, Section 3 introduces the research gaps identified in this study. In Sect. 4, a conceptual framework is proposed explaining how to minimize and control the issues identified as research gaps, whilst Sect. 5 discusses the different technologies and methods used in handling the research gaps. Finally, Section 6 concludes the paper.

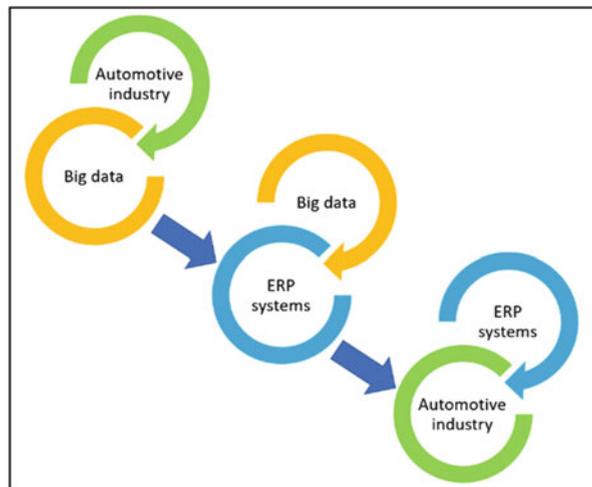
F. Bandara · U. Jayawickrama (✉)
School of Business and Economics, Loughborough University, Loughborough, UK
e-mail: u.jayawickrama@lboro.ac.uk

2 Systematic Literature Review

Choosing papers related to the three areas selected and to find papers with the correlation was a real challenge. Therefore, the research was conducted in three phases, which is represented in Fig. 1. This helps to discover the combinations between the ERP systems and big data, automotive industry and big data and automotive industry and ERP systems. Searching papers were limited to the last 8 years, up to the year 2012, as a measure taken into consideration in collecting the most recent data about the technologies and the combinations.

Figure 2 illustrates the process carried out in the systematic literature review, whilst Table 1 illustrates the statistics related to the SLR. In the initial phase, read the abstract of nearly 30 papers to determine the connections between big data and the automotive industry. In phase 2, 35 papers were read to determine the connection between big data and ERP systems. Finally, in phase 3, 27 papers were taken into consideration in order to determine the combination between ERP systems and the automotive industry. Additionally, six other sources (such as industry papers and trusted websites) were made use of whilst conducting the research. Out of the 30 papers read in phase 1, 10 were omitted reading the abstract as it did not highlight the combination between big data and the automotive industry whilst omitting 11 papers from phase 2 and 13 in phase 3. Out of the 64 papers left, 36 papers were omitted due to relevancy issues and due to information validity issues after reading the content. Finally, cross-referenced 10 papers (applied forward and backward search) were added to the list determining the combination between big data and automotive industry, big data and ERP systems and automotive industry and ERP systems. This includes a conference paper that was found determining the combination of the three areas resulting from a collection of 38 sources for the research.

Fig. 1 Combination used in a systematic literature review



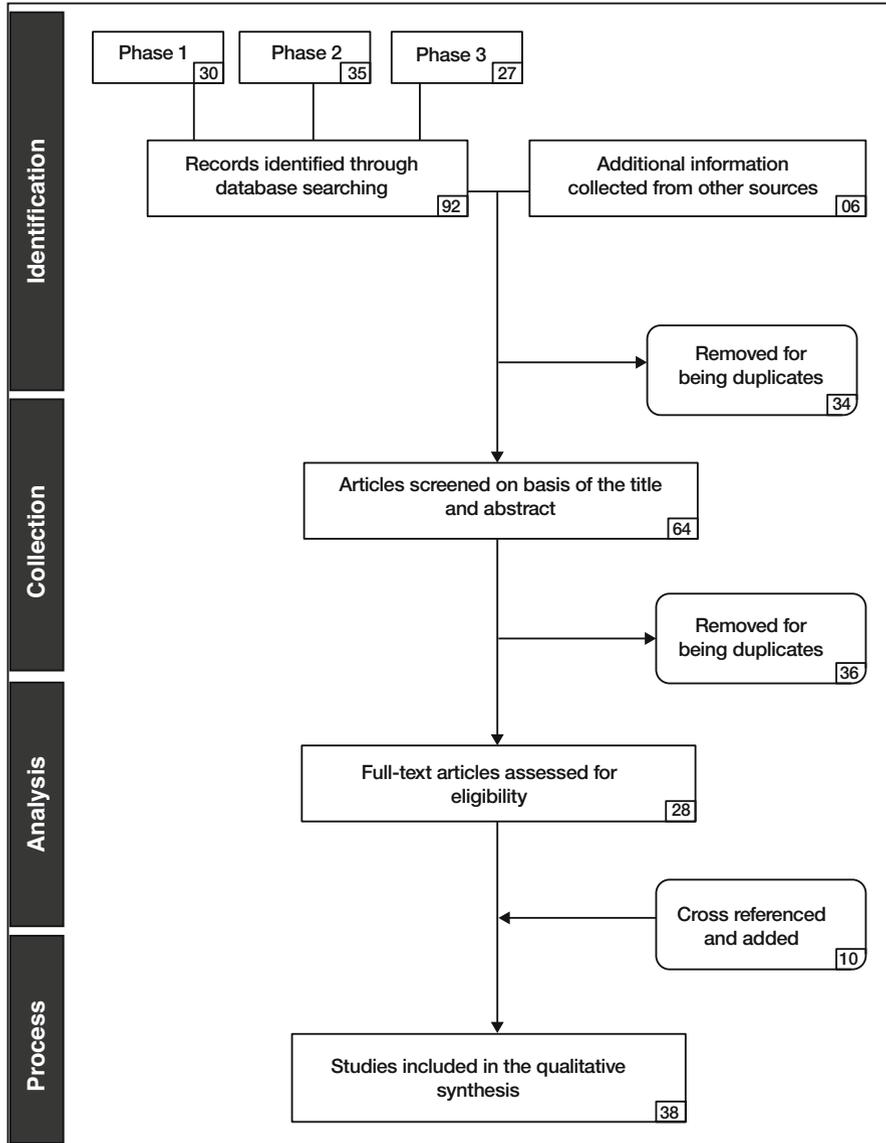


Fig. 2 SLR framework

3 Findings of Systematic Literature Review

ERP systems are one of the most important information systems used in the present business world that can seamlessly integrate different business processes across the departments and functional areas into a centralized system [1–3, 9].

Table 1 Results after conducting the systematic literature review

	Big data and automotive industry	Big data and ERP	ERP and automotive industry	ERP and big data and automotive industry	
Phase 1	30				
Phase 2		35			
Phase 3			27		
Total					92
<i>Addition of other sources</i>	3	1	2		
Total	33	36	29		98
<i>Removed reading the abstract</i>	10	11	13		
Total	23	25	16		64
<i>Removed reading the content</i>	13	14	9		
Total	10	11	7		24
<i>Added later</i>	2	4	3	1	
Full total	12	15	10	1	38

ERP systems are considered as the fabric that connects people, processes, data and things in an intelligent and strategic manner that allows manufacturers to create value from new data streams [7, 10]. Manisha [2] points out the need for an ERP system in the automotive industry in integrating various value chain activities, managing inventory and handling and monitoring several projects simultaneously. ERP systems in the automotive industry rigorously assist in enhancing enterprise visibility and strengthening operational excellence whilst reducing operational errors and improving customer relationship management and real-time information access [2, 11]. The idea of many studies related to ERP systems is largely focused on the implementation of ERP systems [10, 12–14], advantages and disadvantages of implementing ERP systems including the selection of ERP systems for a particular industry [4] or in general [15]. There are relatively very few studies that specifically focus on the implementation of ERP systems and its impact in the automotive industry, whilst the very recently emerged concept of big data in the ERP systems result in a limited number of papers. Nonetheless, ERP is not the same when it comes to the automotive industry as there is a specific set of features to be combined in implementing an ERP system for the automotive industry. Quality and supply chain management, electronic document database, inventory optimization, easy finance management, automobile maintenance, integration with communication channels and built-in original equipment management (OEM) of shipping labels along with electronic data interchange (EDI) templates are some of the features the automobile companies expect to be seen in an ERP system [2, 11, 16].

Table 2 Identified Research gaps and referred papers

Problem	References	ERP	Big Data	Automotive/Automobile Industry
Data management	[2, 5, 10, 16, 18]			X
Trust issues	[2, 5, 10, 18]	x	x	
Complexity of ERP responsiveness	[8, 14, 19]		x	X

The automotive industry is one of the oldest industries since the industrial revolution but manages to withstand the emerging technologies and the development of innovative technologies day by day. The most innovative concept of connected cars, also known as self-driving cars, has resulted in the stimulation of several other technologies such as artificial intelligence, neural networks, machine learning, edge computing and cloud computing. Connected cars are the vehicles connected to always active networks through the convergence of automotive and information technologies [17].

Similarly, big data has become a vital field of research in the area of automotive industry [18]. Big data is referred to the increased volume of data that are difficult to store, process and analyze through traditional database technologies [8, 17, 19, 20]. It has been very recently the researchers have started to link big data with the automotive industry. Emergence of modern concepts such as connected cars is considered as one of the main reasons [17, 18] for the merging of big data with the automotive industry (Table 2).

The next few subsections attempt to provide a clear view of past studies carried out related to the impact of ERP systems and big data domains in the field of automotive, with an intention to provide a theoretical foundation to the integration of ERP systems and big data with the automotive industry. For the ease of understanding and introducing the main aspects of ERP systems for the automotive industry and big data for the field of automotive industry progressively, this section classifies literature into three categories: (a) the general blending of ERP and big data, (b) the integration of big data with the field of automotive and (c) the concept of ERP and its interaction with the automotive industry. Finally, Section 3.4 summarizes the research gaps.

3.1 General Blending of ERP Systems with Big Data

It doesn't matter how technologically advanced and powerful a computerized system that an enterprise has [8]; the impact of big data has become a mandatory fact under the context of collecting and processing large chunks of structured and unstructured data by ERP systems. Komal [8] considers ERP systems as a data bank which is not capable of handling big data. The start of utilizing data from social networking sites by companies as an approach to understand the customer's

behaviour and the use of sensor networks in companies have resulted in the merging of big data to ERP system [3, 4, 8, 14, 20, 21]. The technology of big data does not alter the functionality or the methods used by the ERP system [3, 10]. Blending the two technologies of ERP and big data results in benefitting four areas such as forecasting of sales, scheduling, improving supply chain management and standardizing the practice of hiring [8].

3.2 ERP and Its Influence on Automotive Industry

ERP systems play an increasingly important role in contemporary business technology [8] and are designed to provide seamless integration of processes across functional areas with improved workflow and standardization of various business practices [14]. Lorenc [14] discloses the main potential of ERP systems in the automotive industry is reducing the cost of manufacturing, warehousing, transport with parallel maintenance of efficiency of these processes and technical and operational integration of business functions. Therefore, the ERP systems in the automotive industry assist in harmonizing the information stream with the material flow of goods and services whilst maximizing profitability [16, 17, 22]. Implementation of ERP systems influences the quality and efficiency of processes benefitting immensely by sharing business information, enhancing communication and collaboration, improving the supply chain, improvising customer relationship management, allowing to respond fast to the changing environment, reducing inventories, shortening cycle times, lowering costs, increasing productivity and providing better customer services [14, 23, 24].

3.3 Integration of Big Data with Automotive Industry

The impact of big data in the automotive industry can be categorized into two sections as the impact on the automotive industry and on automobile manufacturing. Despite the fact that in many instances both the names used to represent the same detail, automobile manufacturing refers to the design, development and production of a passenger vehicle to operate on the ordinary road by considering the emerging and innovating technologies, whilst the automotive industry refers to the entire automobile manufacturing process of designing, developing, producing, marketing and selling of motor vehicles [25]; automobile manufacturing is a subset of the automotive industry.

Big data in the automotive industry helps in many areas such as in the concept of connected cars, in providing an automated insight for design and production, predictive maintenance, automated service scheduling after sales, automobile financing, supply chain improvement and vehicle sales marketing [18], whereas the impact of big data in the automobile manufacturing helps in infotainment; the installing

and managing of the collection of hardware and software in automobiles that provides the audio and video entertainment, navigation; the management of global positioning system (GPS), fleet management; and the management of functions that allow companies to rely on transportation business to remove or minimize the risk-associated [26] remote diagnostic, automatic collision notification, enhanced safety, usage-based insurance, traffic management and autonomous driving [18].

It is evident that the main reason behind the usage of big data in the automotive industry is as a result of the usage of sensor technology in automating vehicles which ends up collecting a large amount of unstructured data [5]. Deloitte [5] explains the main reasons for this major shift from human-driven cars to self-driven cars are the shifting of market conditions with the change in the business world. Moreover, arriving of new market entrants resulting in increased competition within the companies, globalization and cost pressure occurring with the fluctuation of currency rates and volatility taking place in the buyer's preference and supplier's choice of selling have resulted in the shift to self-driven cars [27, 28].

3.4 Research Gaps Identified

As a result of conducting the systematic literature review, analyzing the key findings and limitations in the papers selected, research gaps were identified and were categorized into three sections as data management, trust issues and complexity of ERP responsiveness. A conceptual framework has been proposed based on the literature reviewed in Sect. 2. Figure 3 demonstrates the relationship between the two technologies, i.e. ERP systems, big data and the automotive industry, and where the identified research gaps emerge.

Data Management

Data is the most important factor in the fields of the automotive industry and automobile manufacturing as it is the core of every and each decision-making process [6]. Not having a proper data management system worsens business decisions and results in inaccuracy problems [8]. Similarly, management of insufficient data within the company causes many problems internally as well externally [2, 4, 29] by leading to numerous problems in knowing customer's preference in the automotive industry as well as problems in preferences readily available in a short time [14]. Improper data management may cause exceeded data storage by collecting online information and converted to critical data by organizations to understand the customers' behaviour online, which results in data storage exceeding in the ERP systems [14, 30]. Exceeding data storage can affect the processing capacity of data leading to higher profit losses for the company [20]. And the lack of appropriate data management challenges in determining the success of big data strategy in an automotive industry [31].

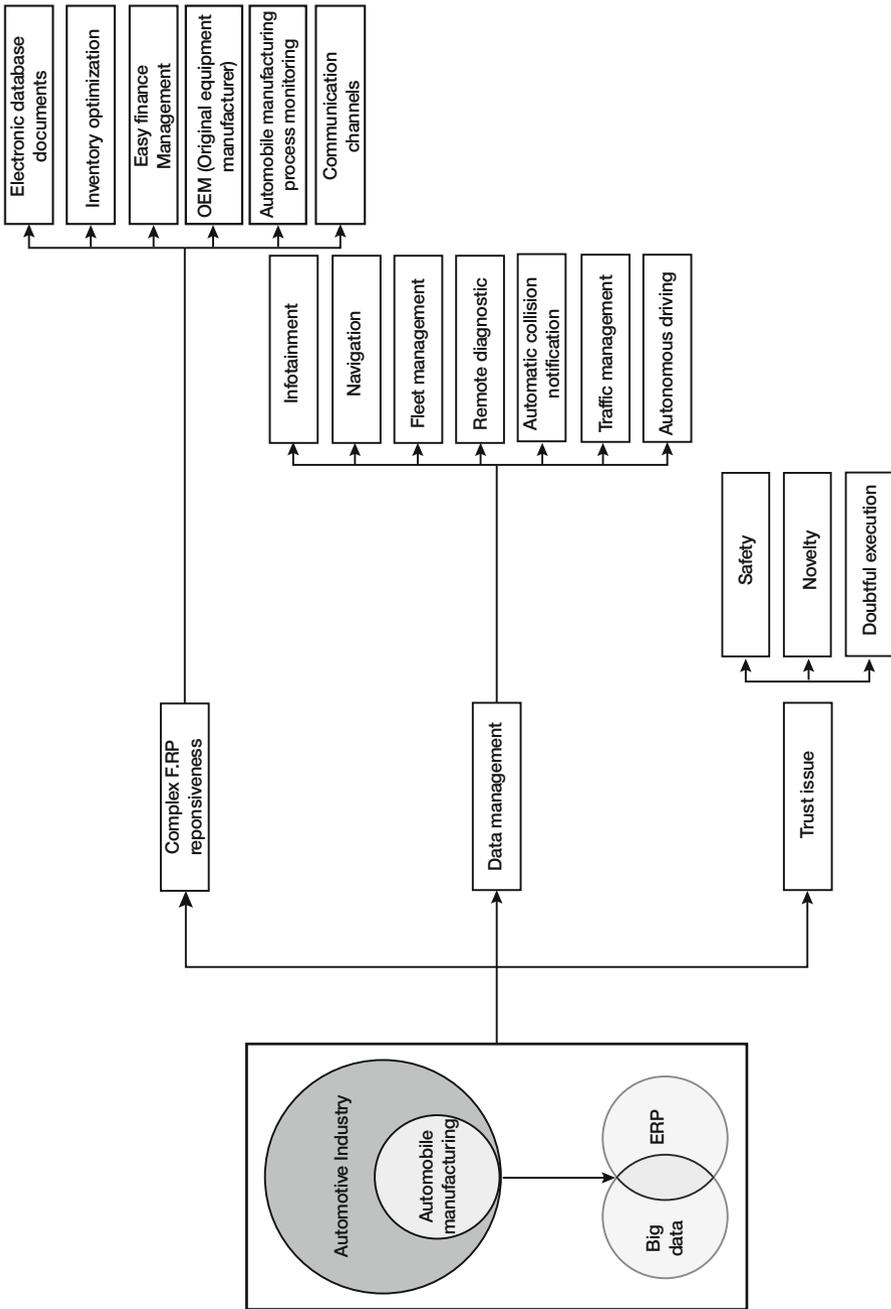


Fig. 3 Research gaps

Trust Issues

Gill [32] illustrates the external factors such as the lack of structure and consistency in the original vehicle manufacturer as well as the internal factors such as the inability to count on detectors that sense the driver's emotions and perceptions causes trust issues among the drivers due to the uncertainty on violation of their privacy and security in the information space. Users do not trust the driver support system [28, 33] for unknown reasons such as being unsure about the safety which is being handled by a device or a program controlled by artificial intelligence [28], the novelty of the technology as the user considers it to be less experienced and the fear of adopting to an automated version of some functions within a vehicle as a cause of ethics and disciplines within a driver [34]. The driver-assist functions are refused by the drivers due to the unavailability of the functions when needed and automatic deactivation [6]. Cruise control that maintains the uniformity of the speed when travelling in motorways and brake assist function that performs the automatic brake by assisting the driver to brake more effectively minimizing the collisions with the vehicle in front and tractions control standards are some of the commonly inbuilt driver-assist functions currently available in a vehicle [28, 33, 35]. Failure of automotive electrical equipment, the sudden failure of sensor-based diagnostic and prognostic, is emphasized as the failure of automotive electrical equipment which can result in fatal consequences [4].

Complexity of ERP Responsiveness

Complexity in the manipulation of data in the ERP systems highly affects its responsiveness and mobility [2, 11, 16, 32]. Gill [32] describes how it demands new levels of collaboration throughout the supply chain, inside and outside the enterprise [36]. Improper management of data as mentioned in Sect. 2 also increases the complexity of ERP responsiveness and reduces the mobility function [1, 13, 20].

4 Development of Conceptual Framework and Discussion

It is vital to select and propose carefully the appropriate solutions for each research gap identified with the intention of bridging them. In this paper, a conceptual framework is proposed as a solution to bridge each category of research gaps shown in Fig. 4.

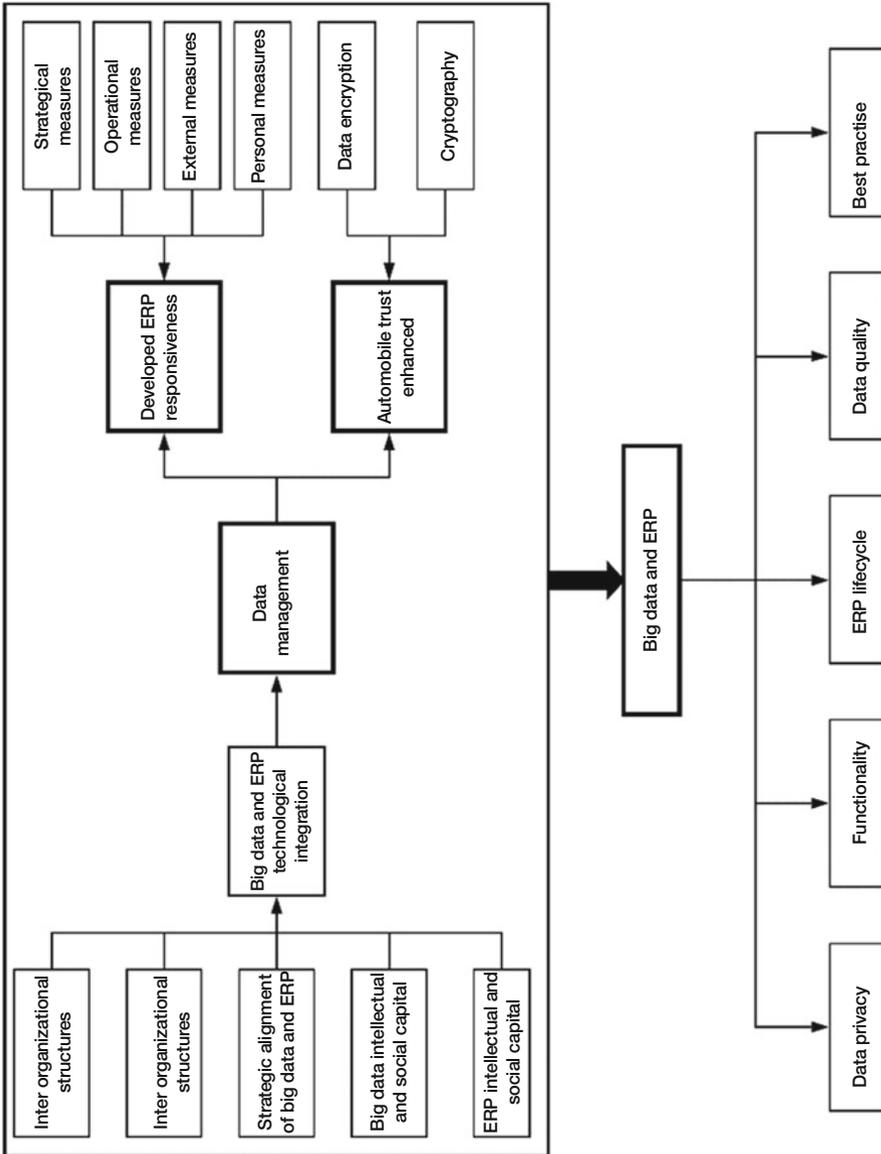


Fig. 4 Conceptual framework: Interactions of ERP systems and big data in the automotive industry

4.1 Solving the Issue of Data Management

Automotive industry with its rapid changes both internally and externally has made a fine combination between its manufacturing and the service divisions. With the development of self-driven cars or the automation of vehicles, it is quite evident that most of the data collected through sensors or other means are big data. Infotainment, navigation, fleet management, remote diagnostics, automatic collision notification, enhanced safety, traffic management and autonomous driving are a result of big data collected in automobiles [18]. Similarly, the big data collected are beneficial in the automotive industry in automating insight for design and production, predictive maintenance, automated service scheduling, making future market predictions, automobile financing and supply chain improvement [18].

Big data indicates an amount of data that is difficult to store, process and analyse using traditional database technologies [19]. Big data collected through the vehicle sensors of the autonomous vehicles are analysed and then used to make various decisions and developments in automobile manufacturing as well as to manage the automotive industry. The analysed data is monitored and used by the ERP system for quality control management which tracks products in real time to notice the problems and improve them ensuring high-quality standards of automobiles and automobile maintenance which gives a comprehensive picture of the product and the equipment status [19]. With the ability to make future predictions in the production of autonomous vehicles, its future market is highly affected in making a number of business decisions that are accurate by maintaining the quality of the entire industry's lifecycle [29].

The strategic alignment, intellectual and social capital integration and technological integration (SIST) model demonstrates a set of guidelines used in managing data along with the transactional data [20]. In order to solve the data management problems regarding the ERP systems is by further enhancing the ERP system according to the data adaptability. Embedding the ERP system with open source big data tools such as the Apache Hadoop to scale up data processing, Apache Spark to process batch and real-time data act as measures to deal with in-memory data processing capabilities, whilst mongo DB acts as a cross-platform compatible source which is considered as one of the most prominent data stores with greater flexibility of configuration with a cloud-native deployment.

4.2 Trust Issues

The concept of energy-efficient intelligent vehicles [37] introduced as a solution for sustainable mobility can be used in order to address the issue of safety by considering the components such as electrical motor system, control strategies and sensor system. With the innovation of self-driven cars and as everything in the self-driven car has taken control through a sensor system [38], it is a major necessity to

tighten the security with the sensor system. Information encryption and decryption methodology can be used as a solution to overcome the trust issues faced by the users.

4.3 Complexity of ERP Responsiveness

In order to solve the complexity of ERP systems in the automotive industry, the concept of four key tenants [36] can be used. The four key tenants are strategic, operational, external and personal measures. Fox [36] clarifies the strategic measures as supporting the management team's ability to access information and respond to changes in the business and the market that allows them to be in greater control. This requires getting the right information in the context of the decisions that need to be made in a form that can be quickly digested and with the ability to take immediate action. Operational measures describe as being able to react to day to day requirements and changes, dynamically managing business activities, automating and optimizing processes and dealing with exceptions accordingly [36]. Operational measures are the backbone of any business, and it demands comprehensive, workflow-enabled, industry-specific, globally-compliant and integrated ERP systems. Fox [36] explains that external measures are about supporting an ongoing collaborative exchange with customers and partners. It is about providing differentiated service levels through every touch point with the organization and being able to identify problems and opportunities whilst empowering the business to act quickly and decisively. Personal measures describe valuing employee's time, leveraging their knowledge and experience, empowering them to manage exceptions whilst embedding and automating best practice processes [36]. Following the four tenants' concept in creating a more responsive ERP system with enhanced mobility will move towards the designing of a much simpler ERP system which is much customized and suitable for the automotive industry. Yet, for instance, SAP Vora can be introduced as an ERP system designed and developed especially to the automotive industry by solving some of the limitations mentioned in Sect. 4.

5 Conclusion

Combination of big data and ERP systems in the automotive industry is a less spoken area where a very limited amount of papers were found in the SLR. Despite the fact that fewer research studies in the combination of the three areas, it is better to recognize the availability of the correlation universally. Analyzing the link between the three emerging areas, i.e. ERP systems, big data and the automotive industry along with its innovative concept of connected cars, it is evident that the core is the big data [4, 15, 28, 33]. Innovations of the automotive industry as well as ERP systems collect large chunks of data where the big data is used during the analysis

process. And on the other hand, it is the centralization of the entire company caused by the robust development of the ERP systems [14, 24, 36]. ERP systems not only are centralizing the operational area of the companies in the automotive industry but also provide a transparent and accurate insight on the functional areas of the company as well [1, 3, 21, 39]. The most common witnessing correlation between the three areas had only one conference paper published. It was hard to frame the research gaps into separate groups due to the lack of evidences proving the relationship of the three areas: ERP systems, big data and the automotive industry.

Further research may focus on determining the straight combination between the self-driving cars and the ERP systems where the neural networks involve in the middle. Despite neural networks [34, 40], other technologies related to self-driving cars such as telematics and machine learning [6, 27] will be researched as a measure of determining the correlation of big data and ERP systems in the automotive industry that is mostly interacting with the innovative automobile concepts.

References

1. U. Jayawickrama, S. Liu, M. Hudson, Empirical evidence of an integrative knowledge competence framework for ERP systems implementation in UK industries. *Comput. Ind.* **82**, 205–223 (2016)
2. M.S., Why does an automotive industry need ERP?, *Quora*, 2018. [Online]. Available: <https://www.quora.com/Why-does-an-automotive-industry-need-ERP>. Accessed 29 Jan 2019
3. U. Jayawickrama, S. Yapa, Factors affecting ERP implementations: Client and consultant perspectives. *J. Enterp. Resour. Plan. Stud.* **2013** (2013)
4. C. Wickman, J. Orlovska, R. Soderberg, Big data usage can be a solution for user behavior evaluation: An automotive industry example. *Procedia CIRP* **72**, 117–122 (2018)
5. Deloitte LLP, Big data and analytics in the automotive industry: Automotive analytics thought piece Contents, p. 16, 2015
6. D. Matthews, Data is key to autonomous vehicle technology, *SmartDataCollective*, 2018. [Online]. Available: <https://www.smartdatacollective.com/data-key-autonomous-vehicle-technology-tesla-says-winning/>. Accessed: 30 Jan 2019
7. U. Jayawickrama, S. Liu, M.H. Smith, P. Akhtar, M. Al Bashir, Knowledge retention in ERP implementations: The context of UK SMEs. *Prod. Plan. Control* **30**(10–12), 1032–1047 (2019)
8. K. Saxena, The future of erp with big data businesses systems and impacting. *Int. J. Adv. Electron. Comput. Sci.* **3**(9), 25–27 (2016)
9. T.H. Davenport, Putting the enterprise into the enterprise system. *Harv. Bus. Rev.*, 121–132 (1998)
10. H.M. Al-Sabri, M. Al-Mashari, A. Chikh, A comparative study and evaluation of ERP reference models in the context of ERP IT-driven. *Bus. Process. Manag. J.* **24**(4), 943–964 (2018)
11. Plex, Must-have ERP features for the automotive industry, *Manufacturing .Net*, 2014. [Online]. Available: <https://www.manufacturing.net/article/2014/01/must-have-erp-features-automotive-industry>. Accessed 30 Jan 2019
12. M. Ali, L. Miller, ERP system implementation in large enterprises – A systematic literature review. *J. Enterp. Inf. Manag.* **30**(4), 666–692 (2017)
13. V. Beal, ERP-enterprise resource planning, *Webopedia*. [Online]. Available: <http://www.webopedia.com/TERM/E/ERP.html>

14. A. Lorenc, Customer logistic service in the automotive industry with the use of the SAP ERP system, *2015 4th Int. Conf. Adv. Logist. Transp.*, pp. 18–23, 2015
15. W. Tsai, P. Lee, Y. Shen, H. Lin, A comprehensive study of the relationship between enterprise resource planning selection criteria and enterprise resource planning system success. *Inf. Manag.* **49**(1), 36–46 (2012)
16. J. Carr, ERP in the auto industry, *Ultra Consultants*, 2016. [Online]. Available: <https://ultraconsultants.com/erp-in-the-auto-industry/>. Accessed: 29 Jan 2019
17. J. Kim, H. Hwangbo, S. Kim, An empirical study on real-time data analytics for connected cars: Sensor-based applications for smart cars. *Int. J. Distrib. Sens. Netw.* **14**(1) (2018)
18. A. Rastogi, Impact of big data on the automotive industry, *newgen apps*, 2018. [Online]. Available: <https://www.newgenapps.com/blog/impact-of-big-data-on-the-automotive-industry>. Accessed 29 Dec 2018
19. I.A.T. Hashem, I. Yaqoob, N.B. Anuar, S. Mokhtar, A. Gani, S. Ullah Khan, The rise of ‘big data’ on cloud computing: Review and open research issues. *Inf. Syst.* **47**, 98–115 (2015)
20. Z. Shi, G. Wang, Integration of big-data ERP and business analytics (BA). *J. High Technol. Manag. Res.* **29**(2), 141–150 (2018)
21. Z. Khan, U. Jayawickrama, P. Akhtar, S.Y. Tarba, The Internet of Things, dynamic data and information processing capabilities and operational agility. *Technol. Forecast. Soc. Change*, 1–32 (2017)
22. K. O’Shaughnessy, Unique industries served by ERP solutions, *Select Hub*, 2019. [Online]. Available: <https://selecthub.com/enterprise-resource-planning/erp-manufacturing-industries/>. Accessed 17 May 2019
23. Techpedia, Database administration, *Techopedia*. [Online]. Available: <https://www.techopedia.com/definition/24080/database-administration>. Accessed 21 Nov 2018
24. D. Threlfall, 7 reasons ERP systems are crucial to automotive industry success, *Worthwhile*, 2018. [Online]. Available: <https://worthwhile.com/blog/2016/10/05/automotive-erp-solutions-roi/>
25. Difference Between, Automotive engineering Vs Automobile engineering, *Difference Between*. [Online]. Available: <http://www.differencebetween.info/difference-between-automotive-and-automobile-engineering>. [Accessed 01 Jan 2019]
26. J. Kim, H. Hwangbo, S. Kim, An empirical study on real-time data analytics for connected cars: Sensor-based applications for smart cars. *Int. J. Distrib. Sens. Netw.* **14**(1) (2018)
27. T. Simon, Massive autonomous vehicle sensor data: what does it mean?, *datanami*, 2017. [Online]. Available: <https://www.datanami.com/2017/05/15/massive-autonomous-vehicle-sensor-data-mean/>. Accessed 2 Feb 2019
28. W.-H. Lin, H. Liu, H.K. Lo, Guest editorial : Big data for driver, vehicle, and system control in ITS. *IEEE Trans. Intell. Transp. Syst.* **17**(6), 1663–1665 (2016)
29. A. Elragal, ERP and Big Data: The Inept Couple. *Procedia Technol.* **16**(February), 242–249 (2015)
30. B. Marr, Big Data in practise, *Bernard Marr & Co.*, 2019. [Online]. Available: <https://www.bernardmarr.com/default.asp?contentID=762>
31. M. Voigt, C. Bennison, M. Hammerschmidt, Gaining traction Big Data in the automotive industry. *Bus. Transform. J.* **10**, 1–2 (2016)
32. S. Gill, Big Data, the Internet of Things, and how ERP can make good on the promise of real-time actionable intelligence, 2017
33. J. S. Apte et al., High-resolution air pollution mapping with Google street view cars: Exploiting Big Data, *Environ. Sci. Technol.*, 2017
34. A. Mylonas, V. Meletiadis, L. Mitrou, D. Gritzalis, Smartphone sensor data as digital evidence. *Comput. Secur.* **38**(2012), 51–75 (2013)
35. T. Orosz, & I. Orosz, Company level big data management, in *SACI 2014 - 9th IEEE Int. Symp. Appl. Comput. Intell. Informatics, Proc.*, pp. 299–303, 2014

36. Malcom Fox, simplifying-erp-reducing-complexity-and-improving-responsiveness-succeed, *Manufacturing.net*, 2015. [Online]. Available: <https://www.manufacturing.net/article/2015/11/simplifying-erp-reducing-complexity-and-improving-responsiveness-succeed>. Accessed 1 Jan 2019
37. I. Bin Aris, R.K.Z. Sahbusdin, A.F.M. Amin, Impacts of IoT and big data to automotive industry, in *2015 10th Asian Control Conf. Emerg. Control Tech. a Sustain. World, ASCC 2015*, (2015), pp. 1–5
38. D. Levinson and P. Investigator, “The Transportation Future s Project: Planning for Technology Change,” 2016.
39. U. Jayawickrama, S. Liu, and M. H. Smith, “Knowledge prioritisation for ERP implementation success,” *Ind. Manag. Data Syst.*, 2017
40. S. Accelerometers, U. Deep, *Estimating Vehicle Movement Direction from Smartphone Accelerometers Using Deep Neural Networks* (Senmsors, 2018)

Software Evaluation Methods to Support B2B Procurement Decisions: An Empirical Study



F. Bodendorf, M. Lutz, and J. Franke

1 Introduction

1.1 Digital Transformation in Procurement

All sectors of the economy and society are currently being influenced by megatrend digitization and its sub-trends. As one example, the automotive industry is also showing the effects of this development [17, 28]. Karl-Thomas Neumann, a former member of the management of Adam Opel GmbH, even speaks of the “biggest change in their history” [17]. The changes within the automotive industry are influenced in particular by the electrification of the drive train, the increasing interconnectedness of vehicles, and the trend of “sharing rather than owning” [4, 11, 13]. Traditionally, purchasing has to master two tasks. On the one hand, contributing to the development of new products and, on the other, managing the overall costs [22]. Digitization reinforces this tension between innovation and cost pressures [10]. As an interface to suppliers, procurement must integrate them efficiently into the innovation process [2, 5]. Fast-moving products that are becoming increasingly digital are taking the center stage of procurement objects [2].

Due to the rapidly progressing digitization, established manufacturers are forced to profoundly change and adapt their products and services as well as the associated business and value creation models [29]. The development of added value, which is characterized by digital components, is increasingly putting traditional

F. Bodendorf (✉) · J. Franke

Institute for Factory Automation and Production Systems, Friedrich-Alexander-University of Erlangen-Nuremberg, Erlangen, Bavaria, Germany
e-mail: frank.bodendorf@faps.fau.de

M. Lutz

TUM School of Management, Technical University of Munich, Munich, Bavaria, Germany

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_63

879

manufacturers under pressure [26]. A 2013 study from the Massachusetts Institute of Technology (MIT) and Capgemini Consulting found that 78% of companies surveyed from different industries believe that digital transformation could be a threat to their business [7]. This shows that most companies are aware of the challenges of digitization. However, over 63% of the respondents stated that their digital adaptation can't compete with the rapid pace of digital transformation [7]. In addition, almost all companies and business units are affected by this issue, as are procurement and its departments. This is particularly true for companies with a high proportion of suppliers' added value. Especially in the field of digital goods and notably as software, it is relatively easy for suppliers to hide the true development effort behind obscure licensing costs [24]. In order to gain transparency in price negotiations, it is therefore of utmost importance for the referring companies to find suitable evaluation approaches [8].

1.2 *Costs of Digital Products*

Due to their characteristic features, digital products have special properties from a cost perspective. Digital goods often have very high fixed costs. The literature in this regard speaks of so-called "first copy costs." These investment costs are also referred to as "sunk costs" because the costs can't be recovered if the product sale fails. Due to the technically simple and cost-effective reproducibility, the variable costs often go to zero [19]. An example of this is a series offered by Netflix, Amazon Prime Video, or similar platforms. Here are the production costs, so the fixed costs or "sunk costs" of the series by spending on actors, set, etc. are very high. The variable costs for the subsequent duplication, which is now completely online and thus requires no additional material products (CDs, DVDs, etc.), are again negligible [21]. This results in the cost characteristics of a material and a digital good shown in Fig. 1. On the left, a cost function of physical goods is shown, in which the production costs increase with increasing production output. This is a typical course, as higher costs of material, a greater number of machine hours, etc. also result in higher costs. On the right hand side is an (ideal) cost function for digital goods. Here, the production costs remain constant with increasing production output, since the variable costs, meaning the marginal costs of production are zero [9, 14].

The unit cost degression can be described as follows: With increasing output, the unit costs decrease because the fixed costs are distributed over a larger quantity. The higher gear the unit cost degression is in, the greater the fixed costs compared to the variable costs are. Due to the very low variable costs (ideal case: zero) for digital goods, the unit cost degression is particularly relevant, and the economies of scale increase with increasing production volume [19]. If the marginal cost rule is adhered to, the digital goods, including software, would have to be offered free of charge, since their marginal costs or variable costs would ideally be zero. This is a purely theoretical train of thought and hardly used in practice. Most software manufacturers today use licensing models to distribute their digital goods. These models invalidate

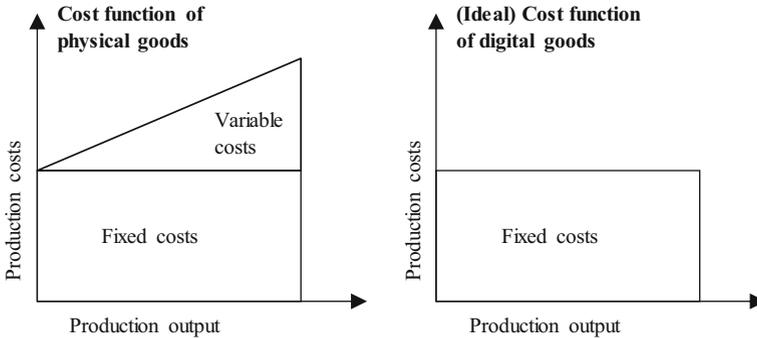


Fig. 1 Cost functions of physical and digital goods [14]

traditional business economics by covering the true cost of manufacturing behind untransparent, license-model-dependent prices.

1.3 Approaches to Software Evaluation

For tangible goods, in contrast to digital ones, there are usually good best-practice approaches that can be used to estimate manufacturing costs with high accuracy because of known material costs, processing steps, etc. For intangibles, especially software, it is very difficult for the software buyer to track an exact value, as the actual development effort is often hidden by the suppliers. Therefore, it is crucial for automotive manufacturers to find viable options for evaluating software components to be procured. Using appropriate new methods of cost evaluation, the price negotiations with suppliers can then be conducted on a substantiated basis of argumentation. However, the goal of these new evaluation approaches is not only to establish a more objective basis for price negotiations but also to ponder how much the new product is worth to the company or the customer [15]. A completely different approach besides the consideration of the costs and the value of a software product is to look at the turnover or profit that is possible or desired on account of the market conditions. This viewpoint, which focuses on the potential success of the product, is reflected in particular in licensing models. In summary, evaluation approaches for checking the plausibility of prices for software to be procured can be divided into three main categories:

Cost-oriented In the case of cost-oriented methods, the aim is to determine or estimate the total development costs of a digital good using suitable procedures. Often, you look at the complexity, the extent, and the resulting expenditure, which must be applied for the development of the software product. On the basis of the calculated effort, the costs can then be deduced or calculated [23]. With cost-oriented pricing, the desired profit margin is then added to the costs. Depending

on whether the perceived value of the product corresponds to the price or not, the customer decides on the purchase.

Value-oriented By contrast, value-oriented methods focus on the customer. Here, for example, the frequently used target costing method determines which benefit the new product will give the customer and what price the customer would be willing to pay. The price is thus determined by the willingness of the customer to pay. Only then the costs are considered, which must lie for a profitable product below the value perceived by the customer, in order to be able to realize a profit margin.

License-oriented Within license-oriented methods, the value and cost of the product are secondary. The license-oriented methods are based on the possible profit or turnover and thus on the success of the product (profit split method). It is estimated, which profits can be achieved with a licensed software. Subsequently, between the software manufacturer (licensor) and the buyer (licensee), for example, the car manufacturer, a profit sharing as a percentage of the sales performance of the final product is agreed. It depends on how much the software has contributed to the success of the product, who bears the risk, how strongly the product is protected against piracy, and many other factors. For example, with a multimedia system in the car, the supplier who developed the software for the multimedia system could claim some of the profits generated by the number of cars sold.

1.4 Research Goal and Design

There are a great variety of methods in the literature for each of the three categories of software evaluation approaches listed. The research questions addressed in this paper are, which of these methods are preferred in corporate practice and what differences are seen between cost-, value-, or license-oriented approaches. These research questions are approached through an empirical study. In this study, on the one hand, data on the use of methods is obtained by means of a questionnaire survey and evaluated quantitatively. On the other hand, interviews with experts provide qualitative insights into the acceptance and effectiveness of the methods used. Figure 2 outlines the study design in the form of a flowchart or process model. The questionnaire design and interview guidelines are described in more detail in Sect. 2. The results of the study provide practical insights and recommendations for software evaluation methods depending on industry and company context. In addition, the study provides insights into the further development of theoretical concepts as well as into future plans of companies.

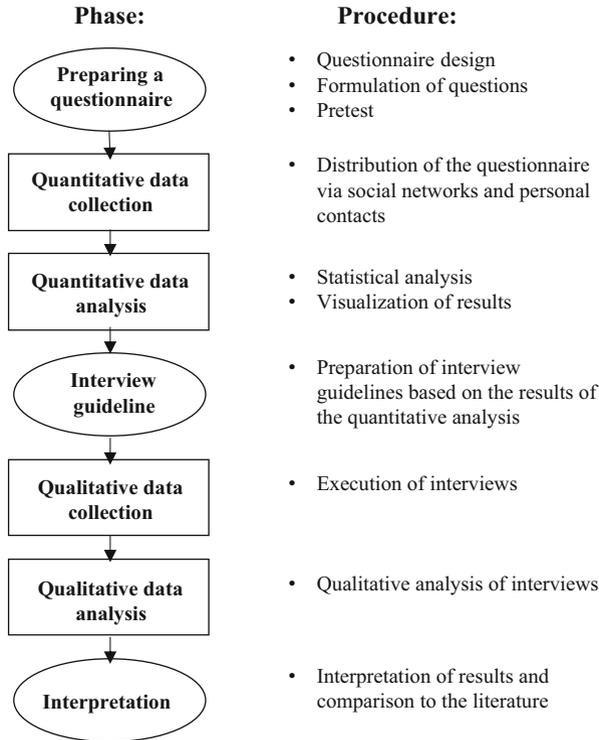


Fig. 2 Study design (explanatory sequential design)

2 Method

2.1 Research Objective

The primary research objective is to investigate which cost-, value-, and license-oriented evaluation approaches for software are described in the literature and which are actually used in current practice. In addition, the evaluation approaches are classified and appropriate evaluation criteria determined. Finally, with the gained findings, recommended actions for business practice can be derived. The research approach of the study is based on the “explanatory sequential design” [3]. This starts with quantitative data collection and evaluation. Building on the results obtained, a qualitative analysis is performed to gain more detailed insights. Thus, with this methodical concept, the evaluation rates for software are first of all theoretically and practically structured in an overview and then further analyzed in detail. This mixed-method approach will answer the research questions listed in Sect. 1.4. The quantitative analysis, which is carried out on the basis of a questionnaire survey, addresses the following research questions:

- What traditional approaches are known in practice to determine software development costs? (RQ1)
- Which methods of software evaluation are currently used by companies in different industries? (RQ2)
- Which nontraditional approaches that are applied in practice are mentioned? (RQ3)

The subsequent qualitative study is based on expert interviews and examines the assessment approaches revealed by the quantitative analysis more precisely by focusing on the following research questions:

- How do experts assess the applicability of the evaluation approaches described in the literature? (RQ4)
- Which traditional evaluation approaches used provide the best results? (RQ5)
- Which nontraditional evaluation approaches are considered most promising? (RQ6)

The course of the empirical quantitative and qualitative analysis is based on the study design presented in Sect. 1.4. A detailed description of the questionnaire and expert interviews follows in Sects .2.3 and 2.4.

2.2 Participant Characteristics and Sampling Procedure

The aim of the quantitative survey is to reach as representative as possible a target group of software procuring companies. The following industries are included:

Automotive, aerospace, electronics, IT/telecommunications, services, mechanical/plant engineering, precision mechanics/optics, biotechnology/pharmacy, commerce/distribution, transport/logistics/transportation, Internet/multimedia.

Differentiation within the branches takes place via the size of the reviewed companies. This is defined by the revenue and number of employees. There is a range from microenterprises employing less than 10 people, with a turnover of less than € two million, to large companies with at least 250 employees or more than € 50 million in sales. The questionnaire addresses persons involved in cost analysis, ideally in a software-related business area. Three different survey channels are used for sample selection:

1. Contacts via research institutes
2. Visits to relevant fairs
3. Professional networks

Most of the participants are gathered over professional networks. In the individual networks, job titles are selected, such as “cost engineer,” “value engineer,” “cost estimation,” “IT buyer,” and other similar job titles. Those who prove to be suitable are contacted directly and asked to participate in the questionnaire. In total, the questionnaire was processed by 47 persons. The answers are checked for

completeness. Ultimately, after the filtering, 36 usable complete data sets remain. The quantitative analysis is followed by expert interviews to look more closely at software evaluation approaches. In addition, the interviews provide estimates of selected and practice-relevant assessment approaches and their evaluation criteria. A total of 36 experts are interviewed. The determination of interview participants is based on the final participants included in the quantitative analysis. At the beginning of the qualitative analysis, the form of the expert interview is determined. Muskat describes three different approaches [16]:

- **Exploratory survey:** This variant is used when there are very little knowledge and experience in a particular subject area. It serves primarily for the first information collection. Often, this method is used as a preparation for the main study in order to then postulate initial hypotheses.
- **Guided expert interviews:** Guided expert interviews systematically ask for a clear target based on specific expertise. This method is used when the research questions are already concrete, and information cannot be obtained from other sources.
- **Plausibility discussions:** This form is used to ensure the relevance of scientific research results and to derive practicable recommendations for action. One approach of this method is to confront the experts with the results of an empirical investigation and to gain an assessment of the findings obtained.

An exploratory study is carried out via a literature analysis and a quantitative online survey (see Sect. 2.3). Through this approach, first helpful insights are gained. Supplementary information on the individual assessment approaches can be obtained from guided expert interviews. In doing so, participants are asked specific questions about different evaluation approaches, which are individually adapted to the respective answers of the questionnaire campaign. In addition, the results of the quantitative analysis are checked for plausibility using expert interviews. The conducted interviews can thus be regarded as a mixture of a guide-based expert interview and a plausibility interview.

2.3 Questionnaire Design

The questionnaire for the quantitative analysis is divided into three sections: characterization of the company, evaluation approaches of software, and evaluation criteria applied.

After the characterization of the company in the first section, it is determined which cost-, value-, and license-oriented evaluation approaches for software are currently used in the company. Seven questions cover which specific procedures the company has in place, which input and output data the procedures require, and which instruments, such as MS Excel or special calculation tools, are used. The last section comprises questions about the evaluation of assessment approaches. In doing so, it is determined whether the company generally assesses the evaluation

approaches pursued and which evaluation criteria are used. Finally, we ask for the average deviation of the cost values estimated from the actual values.

2.4 Interview Design

The interview guideline specifies a certain structure for the expert discussion. The developed guideline consists of three different sections. The first section includes the welcome and presentation as well as the goal of the study. The second section clarifies the corporate and expert context. The third section consists of the technical part of the interview. The name of the company as well as the area of activity and the period of employment of the interviewee are recorded. Afterwards, the evaluation approaches and the evaluation criteria of software are being discussed. In the next step, the expert is confronted with the results of the quantitative survey and asked for his personal feedback. Consequently, a plausibility interview develops from the initially guided expert interview. The aim is to stimulate a professional exchange in which current topics and problems in the field of evaluation of software are addressed and discussed.

3 Results

3.1 Quantitative Results

In the chosen approach of the explanatory sequential design (see Fig. 2), the quantitative evaluation of an online questionnaire survey initially provides an overview of the cross-industry use of software evaluation approaches. Subsequently, the qualitative analysis of an expert survey specifically examines specific aspects of plausibility. The combination of quantitative and qualitative methodology allows a deep screening of software evaluation in procurement. The following sections present significant results from these quantitative and qualitative studies.

The database for the quantitative analysis results from an online questionnaire campaign, which was carried out from mid-July to the end of August 2018. A possible alternative to online questionnaires would be a paper-pencil survey. However, the decision was made on an online questionnaire because it can be distributed to the target groups via e-mail or social media so that a large number of people can be reached without incurring greater financial and time expenditure. In addition, the data are directly available through the electronic input of the respondent, which facilitates later evaluation and minimizes errors.

The first section of the questionnaire asks for the name, industry, and size of the company. This helps to characterize the companies participating in the survey. The quantitative collection, analysis, and evaluation of the survey data are carried out

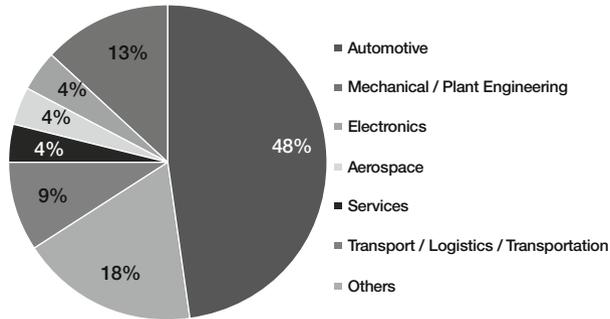


Fig. 3 Allocation of branches out of the questionnaire survey

Table 1 Categorization of companies by size

Description	Number of employees	Revenue
Micro-sized company	≤9	≤2 Mio. EUR
Small-sized company	≤49	≤10 Mio. EUR
Medium-sized company	≤249	≤ 50 Mio. EUR
Large-sized company	>249	>50 Mio EUR

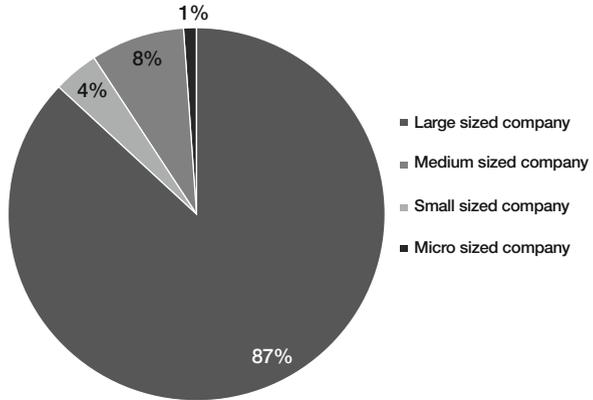
using the software package SPSS. The automotive industry (48%) dominates in the evaluation of the industry reference (see Fig. 3). The second largest industry share can be assigned to the mechanical and plant engineering sector with almost 18%. This puts one focus of the study on the two top-selling industries in Germany. On the one hand, the expert surveys from the automotive, mechanical engineering, and plant engineering sectors provide a broad range of specialist knowledge and, on the other hand, a wide range of opinions on evaluation issues in software procurement so that important insights can also be transferred to other industries.

Within the branches, there is differentiation according to size categories. In order to simplify the classification of the participating companies and to make the category quantifiable, the classification shown in Table 1 is used. The classification is based on the recommendation of the European Commission in the Official Journal of the European Union [6].

Figure 4 illustrates the distribution of company size. At 87%, most of the companies addressed in the survey rank among the big companies. In addition, two micro-enterprises and one medium-sized enterprise are also involved in the survey. In summary, the statements from the subsequent qualitative analysis are based mainly on expert opinions from large companies. As a basis for the opinion on the evaluation of software, the questionnaire contains a given list of theoretically possible evaluation methods, which results from a literature study. Figure 5 shows an overview. The listing serves only as an informative introduction and assistance and can be supplemented by the respondent by additional mentions.

Approximately 75% of recoverable questionnaires confirm the use of one or more of the listed approaches. Afterwards, specific questions will be asked about

Fig. 4 Allocation of companies out of the questionnaire survey



Cost-oriented approaches					
Model-based	COCOMO	Putnam's Model (Slim)	Function Point Analysis	SEER-SEM	
Expert-based	Delphi Method		Work-Breakdown-Structure (WBS)	Planning Poker	
Learning-based	Analogy	Neural Networks	Fuzzy Method	Classification and Regression Trees	
Others	Linear Models	Regression Models	Top-Down	Bottom-Up	

Value-oriented approaches					
Conjoint Analysis	Expert Interviews	Focus Group Interviews	Assessment of Value-in-Use	Ranking Scales	Contingent Evaluation Methods

License-oriented approaches	
Profit Split Method	Knoppe-Formula (25% Rule)

Fig. 5 Evaluation approaches for software products

the individual methods that are currently used in the companies. There are many different mentions for this. The percentage distribution can be seen in Fig. 6.

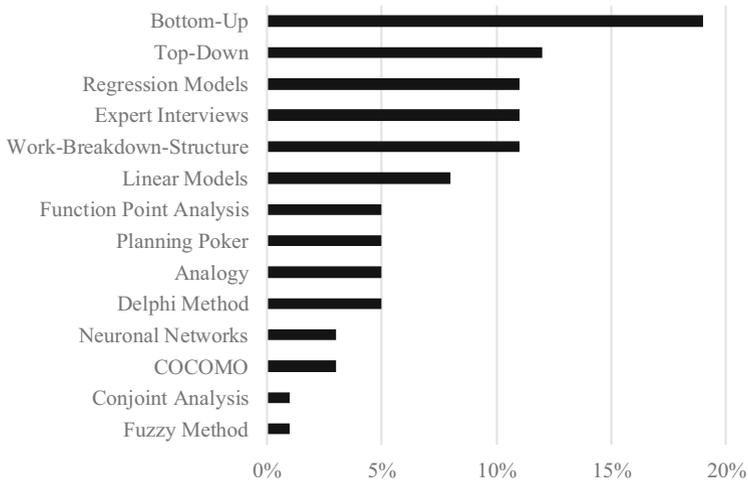


Fig. 6 Allocation of evaluation approaches out of the questionnaire survey

The evaluation results regarding the cost-, value-, and license-oriented evaluation approaches are discussed below.

Cost-oriented approaches The most commonly used methods can be assigned to cost-oriented evaluation approaches. They can be divided into four further subcategories: the model-based, expert-based, learning-based, and “other” methods. It turns out that the model-based methods, which include the COCOMO model, Putnam’s model, and the function point analysis, show relatively little use in practice. This could be related to the fact that the effort and the required expertise for these methods are very high or the methods are no longer applicable to modern software architectures. The methods mentioned are also mostly associated with traditional software manufacturers, hence, the supplier side, in the literature. The expert-based methods, which include the Delphi method, the work breakdown structure, and the planning poker, are more common in practice. Especially work breakdown structures are a very popular approach, which was also confirmed in the later expert interviews [25]. Learning-based methods, which include the analogy method, neural networks, or the fuzzy method, are not strongly represented in practice. The most common one mentioned here is the analogy method. In this case, the costs of the new software to be purchased are deduced from an already performed “old” procurement project. The category “other” covers most of the more commonly used methods. These include the evaluation approaches top-down, bottom-up, regression, and linear models. The upfront designation is the bottom-up approach at 19% and the top-down approach at 12%. However, it should be noted that the top-down and bottom-up approaches are very general and also serve as a basic approach to many more concrete methods. So for example, an analogy method or an artificial neural network may be performed as a top-down or

bottom-up calculation. The regression models are mentioned third with 11% and the linear models with 8%. These two categories include, among others, simple linear regression models, robust regressions, such as least squares estimation, or nonlinear regression models. With 19% combined, these two methods are currently used very frequently in practice.

Value-oriented approaches The value-based evaluation approaches only mention two methods that are listed in the proposal: conjoint analysis with 1% and expert discussions with 11%. With regard to the expert discussions, however, it should be noted that these can be assigned not only to value-oriented but also to cost-oriented methods. Since the online survey only asks for the methods and the respective procedure for these methods is not further specified, it cannot be clearly determined whether the mention of expert discussions as a method is aimed at value-oriented or cost-oriented approaches. However, the subsequent interviews of the qualitative analysis indicate that the method “expert discussions” can be assigned to cost-oriented methods. The conclusion is that value-based evaluation approaches only play a minor role in practice.

License-oriented approaches The two license-based evaluation approaches, the profit-sharing method and the Knoppe formula, are not taken up at all in the questionnaire survey. This confirms the corresponding findings of the literature review. Such methods are not common in practice. Nevertheless, the license orientation is a very interesting approach, which is particularly evident in the later qualitative-oriented expert interviews.

Finally, it is asked for other evaluation approaches that are not listed. About 77% of questionnaire respondents say their company does not use any further evaluation approaches. This result suggests that the structured list presented (see Fig. 5) already provides a broad coverage. If the participants in the online survey indicate that they are planning to use further evaluation approaches, they will then be asked about details. Unfortunately, there are quite a few participants who say that this information cannot be passed because it is confidential. As additional information, only the Bayesian network emerges from the research field of artificial intelligence, more specifically machine learning. From the interviews of the qualitative study, however, a speculative trend for software evaluation methods is discernible. There are many indications that in the future the evaluation of software will be based on approaches from the broad field of artificial intelligence (AI). Methods in this field and in particular in the field of machine learning are currently being tested for their usability and predictive strength.

The quantitative analysis shows which evaluation approaches for software products or components to be procured are currently focused on in practice. In this regard, important insights and perceptions emerge, especially in the automotive and electrical industries, as well as in plant construction. These insights can also be transferred to other areas.

3.2 *Qualitative Results*

In order to obtain supplementary and in-depth information on the individual assessment approaches, six guideline-based expert interviews are conducted. The participants comment on specific questions that are derived individually from the information given in the online survey. Furthermore, the expert interviews serve to validate the plausibility of the quantitative results.

Based on the quantitative evaluation results, the expert survey focuses on methods for cost estimation, utility value analysis, and revenue/earnings analysis. The motivation for this is also supported by a publication of the year 2017. The authors summarize a total of 14 literature reviews including 820 primary studies [20]. Based on the results of this study, the authors set up the following theses on software assessment methods:

- Regression models are dominant.
- New methods are tested in combination, such as analogy methods together with machine learning.
- The use of the function point analysis method is less and less common.
- The trend of the last years points to a great future potential of machine learning.

These theses from the extensive literature analysis are confirmed by the results of the empirical study carried out in a company's context, both as a conclusion of the quantitative evaluation of the online questionnaire survey and as the qualitative findings from the expert interviews.

The questionnaire analysis already shows that regression models play an important role in the evaluation of software (see Fig. 6). This is also confirmed and deepened in the interviews of two experts (Experts 5 and 6). Especially for simple rollover calculations, linear regression models are very popular (Expert 5). Frequently, regression models are flanked by expert opinions and comparisons with former procurement projects. A special approach, "invented" by one of the experts, is the design of a multiple linear regression model for cost drivers. From the identified cost drivers of the software product, the regression parameters and the subsequent determination of the project costs are determined via the model (Expert 5). The use of different combinations of assessment approaches is confirmed in the expert interviews. Among other things, an expert explains a method that represents a combination of the function point analysis and the COCOMO model (Expert 3). In addition, some commonly used methods, such as analogy methods and expert-based methods, are often combined with other approaches, such as machine learning, bottom-up approaches, or regression models (Experts 1 and 5). As found by literature analysis, the function point analysis is currently no longer used in practice. In the online survey, only 5% of participants say they use this method. The interviewed experts confirm this. Only one expert reports on a more intensive examination of function point analysis in combination with the COCOMO model (Expert 3). The trend towards machine learning is confirmed by the experts. Nearly all experts mentioned the importance of this field for the

near future (Experts 1, 2, 4, and 5). In many companies, however, there are only a few competencies in the field of artificial intelligence (Experts 2 and 4). The methods of machine learning generate predictive models by learning from examples or “training data.” Data and empirical values from past procurement projects are indispensable for these methods. However, many companies have inadequately documented and evaluated projects in the past. For this reason, necessary data is missing, with which, for example, artificial neural networks, can be trained. For the years to come, the interviewees see the greatest potential in machine learning methodology combined with selected other approaches shown in Fig. 6. Currently, however, other approaches are in the foreground. The following are some of the experts’ assessments of the practice of evaluation approaches. Model-based assessment approaches, such as COCOMO, Putnam’s model, or SEER-SEM, are considered by experts to be too time-consuming and inefficient due to the lack of knowledge for parameter configuration (Experts 1 and 2). Such parameters are often the lines of code (LOC) or the number of functions of the software under consideration. Two experts explicitly agree that the LOC cannot tell the true size of the software because the LOC depends on many factors, such as programming language, developer experience, and code formatting (Experts 2 and 6). In addition, an approach based on the number of functions is also difficult (Expert 6). More complex software can include up to 70,000 functions today. Breaking it down would take a lot of time. Only a simplified version of the function point analysis would be possible in his opinion, for example, by aggregating about 15 main functions. Overall, the experts agree that virtually all the methods currently in use are based on the integration of expert opinions (Experts 1, 2, 3, 4, 6). Practical and easy-to-use options are the regression models already mentioned as well as the work breakdown structure or the cost breakdown structure, which can be applied either as a bottom-up or top-down approach. The only difference between the work and the cost breakdown structure is that the first one looks at the workload and the last one at the actual cost of software development. Here, the project is broken down into work packages or functions, which are subsequently evaluated by experts individually. These ratings are ultimately aggregated, which then gives the total cost or time required for the software development project [1]. Frequently, software vendors are given such a cost-breakdown or work-breakdown sheets, which they must fill out [18]. In negotiations, the sheets filled out separately by the software buyers and the software providers are compared and the procurement costs or prices negotiated (Experts 3 and 5). However, suppliers are usually in a better negotiating position with this approach, as they have a better basis for argumentation (Expert 3). In addition, it emerges from the discussions that some benchmarking methods are also used. For this, one collects several offers from software providers. Price outliers up and down are filtered out. On the basis of the offers, which are in the midfield, a value range for the costs or the price is determined (Expert 4). The experts are also asked about license-oriented and value-oriented approaches. These approaches are not used at all in the companies of the interviewees. Only one expert mentions the use of the conjoint analysis but admits that the status is “in research” (Expert 3). This is surprising, since value-oriented methods in the field of intangible goods play

a very important role and are widespread. The topic of profit-oriented methods is discussed by the experts with great interest. The main problem is finding a suitable percentage of the software vendor's participation in the software buyer's profit that both the supplier and the customer agree with (Expert 5). For this, among other aspects, you have to determine who bears what risks. For example, if customers do not buy the product, into which the procured software is installed, as assumed, will the software supplier share the loss of revenue? It could not be determined from the expert discussions why, for this reason or for other reasons, license-oriented approaches are not accepted in practice, since it would be a very interesting alternative for both parties (buyers and suppliers).

4 Conclusions

The participants of the questionnaire campaign are experts in the field of software procurement in companies of different sizes and from different industries.

For the returns (47) and the datasets which can be used (36), it is noticeable that the answers which are suitable for further analysis come mainly from large enterprises (87%) in the automotive, plant, and electrical industries (75%). You can only speculate about the reasons. For example, it could be that methodological approaches for software evaluation are not of interest in smaller companies and in other industries or that software procurement itself plays a minor role there. The major results from the quantitative and qualitative analyses are presented in Sects. 3.1 and 3.2 in some details. Figure 7 focuses on method-based key findings derived from those results in the form of six theses.

In an overall view, the research questions of Sect. 2.1 can be answered:

- RQ1: In practice, the best-known software evaluation approaches are function point, COCOMO, expert-, learning-, and regression-based methods.
- RQ2: The most frequently used approaches are expert workshops and value-oriented methods.
- RQ3: Nontraditional methods, as from the field of machine learning, get more and more known in practice but are not applied yet.
- RQ4: Cost-oriented methods are considered deployable in practice, whereas value- and license-oriented methods are seen as an academic playground.
- RQ5: Bottom-up approaches provide best practice costs and a high level of transparency.
- RQ6: The highest potential for predictive cost analytics in the future and particularly for software cost estimation is assigned to machine learning models like artificial neuronal networks.

The qualitative and quantitative analyses are based on a database, which results from the opinions of experienced persons or experts in companies. The findings of this study are to be seen under the light of this limitation with regard to the interviewed stakeholders (Fig. 3). The analysis is carried out from the perspective

Method-centered conclusion	Quantitative argument	Qualitative argument
<p>Cost-oriented methods that are based on mathematical estimation models rarely accepted and outdated.</p>	<p>With 8% of the answers, this approach is not widely applied and is limited to the Function Point and COCOMO estimation models.</p>	<p>Model-based approaches such as COCOMO are time-consuming to implement. COCOMO and Function Point are no longer up-to-date. Their use is decreasing, however, some combinations are noticed</p>
<p>Cost-oriented approaches that are based on expert-workshops are more appreciated and forestall information asymmetry between supplier and demander.</p>	<p>With 21% of the answers, expert-based methods are used relatively frequently in practice.</p>	<p>Preference is given to using Work- and Cost-Breakdown-Structure approaches both on the supply and demand side . In addition, benchmarking is used to determine value limits for software prices and costs.</p>
<p>Cost-oriented approaches that are based on learning-algorithms can seldom be found but have a high potential given "big data".</p>	<p>Relevant 9% of the responses refer exclusively to Analogy Models and Artificial Neural Networks.</p>	<p>There is a trend towards machine learning, often combining such approaches with other assessment methods. An impediment to more current use is seen in the lack of sufficient training data.</p>
<p>Looking at Cost-oriented methods that are not model-, expert- or learning-based regression methods are dominant by far.</p>	<p>Neglecting the general approaches of Bottom-Up and Top-Down, which are often referred to in conjunction with other methods, the responses focus on regression models.</p>	<p>Regression models are often combined with expert-based approaches. Often, multiple regression is used to model more complex cost structures.</p>
<p>Value-oriented methods are extensively cited in literature but hardly addressed in practice and rather seen as an academic background.</p>	<p>This relatively broad spectrum only includes 12% of the answers, which are solely related to Conjoint Analysis and expert workshops.</p>	<p>The Conjoint Analysis is seen rather as a research-related construct and used for accompanying pilot tests.</p>
<p>License-oriented methods are not seized at all but attract extremely high attention and have a promising future.</p>	<p>There are no entries.</p>	<p>There is great interest in these approaches. The idea of profit split gains attentiveness. . The lack of practical experience is attributed to a lack of awareness.</p>

Fig. 7 Key results

of the software requester respectively from the point of view of the OEM or end-product manufacturers, who usually do not produce software themselves, but install procured software products and software components in their products. Although the majority of the companies surveyed use the bottom-up and top-down approach as a basis, they try to substantiate this with further concepts. The resulting methodology, however, remains at a blurred level, such as the “analogy method” or “expert discussions.” For a more detailed specification, e.g., as the analogy method, a detailed documentation of the data basis and the evaluation are of great importance. Work breakdown structures or cost breakdown structures are widespread procedures that can be subordinated to the bottom-up or the top-down approach. However, this does not lead to more detailed and standardized methods which can be supported by software tools or partially automated. About 58% of study participants use spreadsheets as part of their software assessment, followed by proprietary software (26%) and custom software at 16%. Model-based methods such as COCOMO, Putnam’s model (SLIM), or SEER-SEM are unlikely to meet with demand in practice according to this study. In the study reported here, value-oriented and license-oriented methods are also addressed, but these are currently not used in practice, although a high interest exists.

In summary, it can be seen that there is no standardized method and procedure for evaluating software products to be purchased in terms of costs. Companies derive their approaches from well-known and proven methods for the evaluation of tangible goods. It has to be shown whether specific evaluation methods for intangible goods, especially software, will prevail in the near future. It could also be crucial to generally rethink the cost analysis in software purchasing. Agile software development methods make traditional evaluation approaches more and more obsolete. As a consequence, there may be a stronger trend towards value-oriented and license-oriented approaches. Cost-oriented approaches are moving away from manual calculation and approaching the world of automated learning algorithms from artificial intelligence. The findings of this research undergird the statement that the way of purchasing digital components has changed from pure procurement to value creation [12, 27]. This is especially true for software products. Methodological approaches to support procurement decisions are changing over from traditional isolated cost analysis to a mixed-method approach comprising cost-, value-, and license-oriented techniques. This trend is qualitatively and quantitatively confirmed by the findings of an empirical study covering diverse branches of industry. The study exhibits that out of the great variety of methods to be found in literature, only a few are applied in practice. Popular ones are, for example, regression models and work breakdown structures. However, the empirical study also shows a trend towards intelligent calculation methods based on the analysis of historical data. There is a tremendous interest in machine learning models. The study shows that particularly in German key industries, e.g., automotive, mechanical, and plant engineering, artificial intelligence algorithms and tools are tested to analyze cost structures of software products to be purchased. The first experiences gained are promising. However, there is a lot of work still to be done in order to provide ample consolidated training data coming from different sources. Big companies

are actually investing in building big data platforms or “data lakes” which can be used for software cost analytics among many other applications. From annual reports and industry studies, it can be seen that software companies’ EBIT margins are significantly higher than the average margins in other industries. This shows that software buyers need cost transparency in purchasing. The experts in software procurement interviewed also urgently desire this transparency. The investigation shows that there are no standardized and practically dominant approaches for the evaluation of software products. However, the plausibility of software prices and costs is a key issue in corporate sourcing departments, which is steadily worsening as the digital transformation progresses. Sustainable and effective methods for evaluating software products or components are essential in procurement practice. The study shows that there is still a lot of catching up to do here.

References

1. T. Alby, S. Pflieger, L. Tran. Work Breakdown Structure (WBS)/Projektstrukturplan. In Loox GmbH [online]. <http://projektmanagement-definitionen.de/glossar/work-breakdown-structure-wbs-projektstrukturplan/>. Retrieved [06-08-2020]
2. F. Bienhaus, A. Haddud, Procurement 4.0: factors influencing the digitisation of procurement and supply chains. *Bus. Process Manag. J.* **24**(4), 965–984 (2018)
3. J.W. Creswell, C. Plano, L. Vicki, *Designing and conducting mixed methods research*, 2nd edn. (Sage Publications Inc, Los Angeles, 2011), pp. 68–69
4. W. Diez, *Wohin steuert die deutsche Automobilindustrie?* 2nd edn. (De Gruyter Oldenbourg, Boston, 2018)
5. F. Dinnessen, S. Hellmann. Die erfolgsentscheidende Rolle des Einkaufs in der digitalen Transformation. [online]. <https://beschaffung-aktuell.industrie.de/supply-chain-management/die-erfolgsentscheidende-rolle-des-einkaufs-in-der-digitalen-transformation/#slider-intro-22> (2016). Retrieved [06-08-2020]
6. Europäische Kommission: Empfehlung der Kommission vom 6. Mai 2003 betreffend die Definition der Kleinunternehmen sowie der kleinen und mittleren Unternehmen. Amtsblatt der Europäischen Kommission. Brüssel, 3–4 (2003)
7. M. Fitzgerald, N. Kruschwitz, D. Bonnet, M. Welch. *Embracing Digital Technology. A New Strategic Imperative. MIT Sloan management review in collaboration with capgemini consulting*. (Massachusetts Institute of Technology, 2013), pp. 1–2
8. A. Hoffjan, S. Lührs, A. Kolburg, Cost transparency in supply chains: Demystification of the cooperation tenet. *Schmalenbach Bus. Rev.* **63**(3), 230–251 (2011)
9. K.W. Huang, A. Sundararajan, Pricing digital goods: Discontinuous costs and shared infrastructure. *Inf. Syst. Res.* **22**(4), 721–738 (2011)
10. Z. Kostic, Innovation and Digital Transformation as a Competition Catalyst. *Ekonomika. J. Econ. Theory Pract. Soc. Issues* **64**(1350-2019-2780), 13–24 (2018)
11. S. Koushik, R. Mehl. The automotive industry as a digital business. Management Summary NTT Group (2015)
12. G. Lechner, Contribution of supplier management to company value development. *Eurasian J. Bus. Manag.* **7**(2), 38–48 (2019)
13. C. Legner, T. Eymann, T. Hess, C. Matt, T. Böhmman, P. Drews, et al., Digitalization: Opportunity and challenge for the business and information systems engineering community. *Bus. Inf. Syst. Eng.* **59**(4), 301–308 (2017)
14. J.M. Leimeister, *Einführung in die Wirtschaftsinformatik* (Springer Gabler, Berlin, 2015)

15. H. Li, Y. Wang, R. Yin, T.J. Kull, T.Y. Choi, Target pricing: Demandside versus supply-side approaches. *Int. J. Prod. Econ.* **136**(1), 172–184 (2012)
16. M. Muskat, D.A. Blackman, B. Muskat, Mixed methods: Combining expert interviews, cross-impact analysis and scenario development. *Electron. J. Bus. Res. Methods* **10**(1), 09–21 (2012)
17. K.-T. Neumann, “Achtung, ‘‘Umparker’’! Vom Automobilhersteller zum vernetzten Mobilitätsanbieter”, in *CSR und Digitalisierung. Der digitale Wandel als Chance und Herausforderung für Wirtschaft und Gesellschaft*, ed. by A. Hildebrandt, W. Landhäußer, (Springer, Berlin, Heidelberg, 2017), pp. 373–389
18. U. Nilsson. Product costing in interorganizational relationships: A supplier’s perspective; JIBS dissertation series, Jönköping International Business School (2004), p. 19
19. M. Peitz, P. Waelbroeck, Piracy of digital products: A critical review of the theoretical literature. *Inf. Econ. Policy* **18**(4), 449–476 (2006)
20. S.P. Pillai, S.D. Madhukumar, T. Radharamanan. Consolidating evidence based studies in software cost/effort estimation — A tertiary study; TENCON (Hg.): TENCON 2017–2017 IEEE Region 10 Conference. In cooperation with Norliza Mohd Noor. TENCON 2017–2017 IEEE Region 10 Conference. Penang. 11/5/2017–11/8/2017. (IEEE, Piscataway, NJ, 2017). pp. 833–838
21. T. Rayna, Understanding the challenges of the digital economy: The nature of digital goods. *Commun. Strateg.* **71**, 13–16 (2008)
22. H. Schiele, Early supplier integration: The dual role of purchasing in new product development. *R&D Manag.* **40**(2), 138–153 (2010)
23. G. Schuh, D. Guo, M. Hoppe, V. Ünlü, Steuerung der Lieferantenbasis, in *VDI-Buch. Einkaufsmanagement: Handbuch Produktion und Management 7*, ed. by G. Schuh, 2nd edn., (Springer Vieweg, Berlin, 2014), pp. 255–342
24. Y. Tan, J.E. Carrillo, H.K. Cheng, The agency model for digital goods. *Decis. Sci.* **47**(4), 628–660 (2016)
25. R.C. Tausworthe, The work breakdown structure in software project management. *J. Syst. Softw.* **1**, 181–186 (1979)
26. N. Urbach, F. Ahlemann, *IT-Management im Zeitalter der Digitalisierung. Auf dem Weg zur IT-Organisation der Zukunft* (Springer Berlin Heidelberg, Berlin, 2016), pp. 1–2
27. H. Wildemann. Einkaufspotentialanalyse: Programme zur partnerschaftlichen Erschließung von Rationalisierungspotentialen. (TCW Transfer Centrum GmbH & Co. KG, 2008), pp. 2
28. U. Winkelhake, Winkelhake, & Schilgerius, in *Digital Transformation of the Automotive Industry*, (Springer International Publishing AG, Cham, 2018)
29. C. Zott, R. Amit, Model innovation: How to create value in a digital world. *Mark. Intell. Rev.* **9**(1), 18–23 (2017)

Sentiment Analysis of Product Reviews on Social Media



Velam Thanu and David Yoon

1 Introduction

In the world of e-commerce, product reviews are of great importance to both buyers and sellers. Most of the products are bought online, and reviews are one of the most important and easy ways for a customer to gain trust in the product. From the sellers' point of view, they need to know the performance of their products. The feedback of their customers is crucial to improve quality and service.

There are many ways to gather product reviews (like looking at the review section on the e-commerce website where the product is sold or having a portal for customer issues). But another important platform wherein product reviews pour in is social media. Social media is increasingly used by humans to express their feelings and opinions in the form of short text messages to reach a large audience. Hence, it is important for both the buyer and seller to find a way to analyze people's comments about products on a social media platform.

As there is an immense amount of data available on social media, it is impossible to analyze them manually in real time. We need tools that can do this for us and give an analyzed output. In this project, we have developed an application which gives the sentiment of tweets about the product.

V. Thanu · D. Yoon (✉)

CIS Department, University of Michigan – Dearborn, Dearborn, MI, USA

e-mail: dhyoon@umich.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_64

899

1.1 Purpose

The main purpose of this application is to give a report on the number of positive, negative, and neutral tweets on Twitter posted by humans around the world for any product. It is very simple to use. The user must just enter the name of the product and submit it, and this application gives a graph as an output which shows the percentages of positive, negative, and neutral tweets. It also shows a few sample tweets under the positive and negative categories.

Hence, the user can get a good idea of the latest sentiments of people on a product which they tweeted without spending any time to go through the tweets. In fact, the user need not even have a Twitter account.

1.2 Motivation

There is an immense amount of valuable data available on social media in the form of product reviews. It is important to analyze them for the benefit of both the buyer and seller. For this, we need good tools which can analyze data and give results about the sentiment of users for different products.

The attempt made in this project is to create an application that analyzes the tweets of a product on Twitter and gives the sentiment analysis result. This is highly automated and needs very little effort from the user. From just a click, the user can get valuable data which can impact their buying or selling decisions having known that social media is the platform that is used immensely to express oneself in recent times.

1.3 Brief Description

The application is basically a tool that gives the user the sentiment analysis of products.

There is an immense amount of valuable data available on social media in the form of product reviews. It is important to analyze them for the benefit of both the buyer and seller. For this, we need good tools which can analyze data and give results about the sentiment of users for different products.

The attempt made in this project is to create an application that analyzes the tweets of a product on Twitter and gives the sentiment analysis result. This is highly automated and needs very little effort from the user. From just a click, the user can get valuable data which can impact their buying or selling decisions having known that social media is the platform that is used immensely to express oneself in recent times.

2 Technical Specification

2.1 System Architecture Diagram

Below is the system architecture design diagram of the application. From the front-end webpage, the user submits the keyword/product name. The Flask server receives this keyword and passes it to the sentiment analysis server. The sentiment analysis server is the Python script that is the heart of this application.

It uses the Tweepy library that enables this Python script to communicate with the Twitter API to authenticate and get the tweets based on search keywords. It then uses the TextBlob library to perform sentiment analysis on the tweets. The sentiment analysis results are now passed to the Flask server which displays them on the front-end web page (Fig. 1).

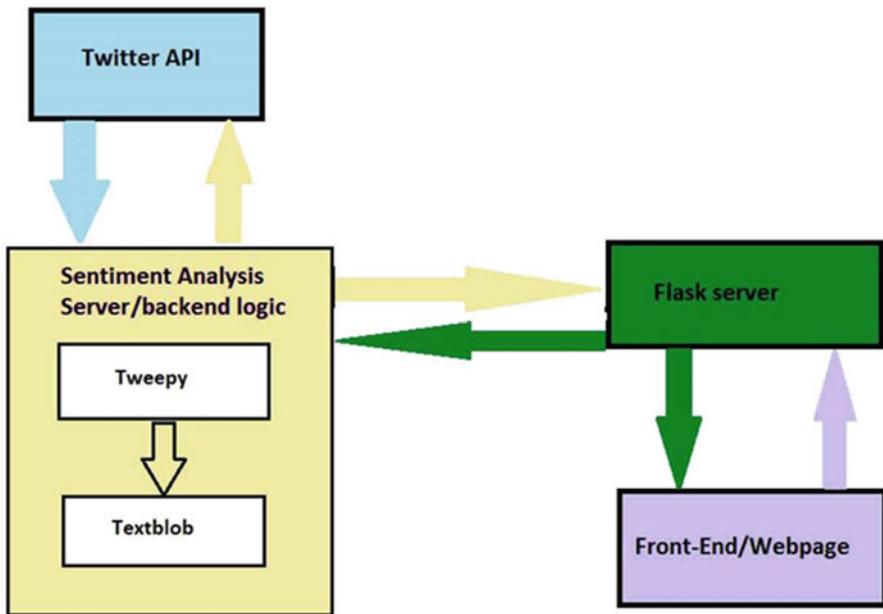


Fig. 1 System Architecture

2.2 Description Libraries and Web Framework Used

Twitter

Twitter is an online news and social networking platform where people communicate and express their feelings in short messages called *tweets*. Twitter restricts every tweet to *280 characters*, which keeps the tweets short and scan-friendly. Users can get the crux of the tweet by just giving a glance which makes Twitter very popular in today's attention-deficit world (Fig. 2).

There are about 600 million daily searches and 10 billion tweets on Twitter. Twitter and its third-party apps offer a way for the users to stay in the loop on what people are saying about a company or brand, directly or indirectly. Twitter helps in putting a face to a company and functions as a customer service platform to many companies.

Due to its openness in sharing and the immense amount of data, Twitter is a prime example of social media in which users can mine interesting patterns and build real-world applications. Sentiment analysis of tweets gives real-time information for disaster relief, using Twitter analytics for improving businesses.

Twitter APIs

An application program interface (API) is a set of routines, protocols, and tools for building software applications. An API makes it easier to develop a program by providing all the building blocks which a programmer puts together to develop a new application altogether.

Twitter allows access to parts of its service via APIs to allow people to build software that integrates with Twitter, like a solution that helps a company.

One of the APIs is the standard search API which is used in this project that returns a collection of relevant tweets matching a specified query.

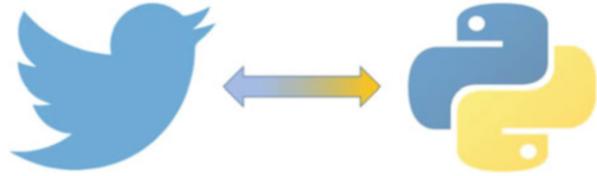
I have used the following two parameters of this API:

Search(q, count) where

Fig. 2 Twitter



Fig. 3 Tweepy



q is A UTF-8, URL-encoded search query of 500 characters maximum, including operators.

count is the number of tweets to return per page, up to a maximum of 100.

2.3 Tweepy

Tweepy is a Python library for accessing the Twitter API (Fig. 3).

It is open-sourced and hosted on GitHub and enables Python to communicate with the Twitter platform and use its API.

TextBlob

Sentiment analysis employs natural language processing, text analysis, computational linguistics, and biometrics to systematically identify, extract, quantify, and study affective states and subjective information. Sentiment analysis is widely applied to the voice of the customer materials such as reviews and survey responses, online and social media, and healthcare materials for applications that range from marketing to customer service to clinical medicine.

In recent years, sentiment analysis has become a hot-trend topic of scientific and market research in the field of natural language processing (NLP) and machine learning. In this project, the sentiments of tweets are analyzed as either positive, negative, or neutral.

For this, we have used TextBlob, which is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun-phrase extraction, *sentiment analysis*, classification, translation, and more.

We are more concerned about the sentiment analysis functionality of the API. The sentiment property returns a named tuple of the form Sentiment (polarity, subjectivity). The polarity score is afloat within the range $[-1.0, 1.0]$.

Looking into the source code of *textblob.en.sentiments* (which is the sentiment analysis module of TextBlob), we get to know that it has a training set with preclassified movie reviews. When we give a new text for analysis, it uses the *Naïve Bayes classifier* to classify the new text's polarity as positive or negative (Fig. 4).

```

class NaiveBayesAnalyzer(BaseSentimentAnalyzer): [docs]
    """Naive Bayes analyzer that is trained on a dataset of movie reviews.
    Returns results as a named tuple of the form:
    ``Sentiment(classification, p_pos, p_neg)``

    :param callable feature_extractor: Function that returns a dictionary of
        features, given a list of words.
    """

    kind = DISCRETE
    #: Return type declaration
    RETURN_TYPE = namedtuple('Sentiment', ['classification', 'p_pos', 'p_neg'])

    def __init__(self, feature_extractor=_default_feature_extractor):
        super(NaiveBayesAnalyzer, self).__init__()
        self._classifier = None
        self.feature_extractor = feature_extractor

    @requires_nltk_corpus [docs]
    def train(self):
        """Train the Naive Bayes classifier on the movie review corpus."""
        super(NaiveBayesAnalyzer, self).train()
        neg_ids = nltk.corpus.movie_reviews.fileids('neg')
        pos_ids = nltk.corpus.movie_reviews.fileids('pos')
        neg_feats = [(self.feature_extractor(
            nltk.corpus.movie_reviews.words(fileids=[f])), 'neg') for f in neg_ids]
        pos_feats = [(self.feature_extractor(
            nltk.corpus.movie_reviews.words(fileids=[f])), 'pos') for f in pos_ids]
        train_data = neg_feats + pos_feats
        self._classifier = nltk.classify.NaiveBayesClassifier.train(train_data)

    def analyze(self, text): [docs]
        """Return the sentiment as a named tuple of the form:
        ``Sentiment(classification, p_pos, p_neg)``
        """
        # Lazily train the classifier
        super(NaiveBayesAnalyzer, self).analyze(text)
        tokens = word_tokenize(text, include_punc=False)
        filtered = (t.lower() for t in tokens if len(t) >= 3)
        feats = self.feature_extractor(filtered)
        prob_dist = self._classifier.prob_classify(feats)
        return self.RETURN_TYPE(
            classification=prob_dist.max(),
            p_pos=prob_dist.prob('pos'),
            p_neg=prob_dist.prob("neg")
        )

```

Fig. 4 TextBlob

Naïve Bayes Classifier

It is a classification technique based on Bayes' theorem with an assumption of independence among predictors.

Bayes' theorem is stated as:

$$P(y|X) = \frac{P(X|y) P(y)}{P(X)}$$

where X is a problem vector

$$X = (x_1, x_2, x_3, \dots, x_n)$$

$P(y|X)$ is the probability of hypothesis y given the vector x . This is called the posterior probability.

$P(X|y)$ is the probability of vector X given that the hypothesis y was true.

$P(y)$ is the probability of hypothesis y being true (regardless of X). This is called the prior probability of y .

$P(X)$ is the probability of the vector (regardless of the y).

The naïve assumption of independence among predictors is now applied, and we can come up with the below Naïve Bayes theorem.

$$P(y|x_1, \dots, x_n) \propto P(y) \prod_{i=1}^n P(x_i|y)$$

Due to good results in multiclass problems and independence rules, Naïve Bayes classifiers are mostly used in text classification and have a higher success rate as compared to other algorithms. As a result, it is widely used in social media sentiment analysis, to identify positive and negative customer sentiments just like it's used in TextBlob.

- In TextBlob, they have first trained the analyzer using a movie review dataset. An overview of how they could have trained is explained below.
- The reviews are preprocessed as punctuations, special characters, etc. and are not needed for sentiment analysis.
- The next step is creating a list of all the words we have in the training set, breaking it into word features. Word features are basically a list of distinct words, each of which has its frequency (number of occurrences in the set) as a key.
- The next step is to go through all the words in the training set, comparing every word against the review at hand and giving a label of 1 if it is present in the review or 0 if it is not present in the review. Also, we can give a positive or negative label for each review. We thus create a matrix of values which are.
- 0 or 1 and positive or negative, respectively, which is called the feature
- Vector.

Fig. 5 Flask



- After creating the feature vector, the `NaiveBayesClassifier.train()` function in python trains the analyzer, and the probability model is created.
- Next, if we pass a review from testing data, it will analyze the review and give us an output of whether it has a positive, negative, or neutral sentiment.

Flask

Flask is a *web application framework* written in Python.

Web application framework or simply Web framework represents a collection of libraries and modules that enable a web application developer to write applications without having to bother about low-level details such as protocols and thread management (Fig. 5).

Flask is developed by Armin Ronacher, who leads an international group of Python enthusiasts named Pocco. Flask is based on the Werkzeug WSGI toolkit and *Jinja2 template* engine. Web Server Gateway Interface (WSGI) has been adopted as a standard for Python web application development. WSGI is a specification for a universal interface between the web server and the web applications. Werkzeug is a WSGI toolkit, which implements requests, response objects, and other utility functions. This enables building a web framework on top of it. The Flask framework uses Werkzeug as one of its bases.

Jinja2 is a popular templating engine for Python. A web templating system combines a template with a certain data source to render dynamic web pages. By convention, templates are stored in subdirectories within the application's Python source tree, with the name `templates`.

In this project, I have used Flask with templates to create a dynamic web page that can take user inputs, process them using the backend python code (sentiment analyzer), and then publish the output from it dynamically on the webpage.

3 Conclusion

To conclude, in this project, the sentiment analysis of product tweets on Twitter was performed successfully, and the results were displayed to the user. The sentiment

analysis was performed using libraries available in python. Also, APIs provided by Twitter were used to communicate and fetch data.

In this way, the immense amount of data available on social media platforms like Twitter can be put to constructive use.

Research on Efficient and Fuzzy Matching Algorithm in Information Dissemination System



Qinwen Zuo, Fred Wu, Fei Yan, Shaofei Lu, Colmenares-diaz Eduardo, and Junbin Liang

1 Introduction

In recent years, with the rapid development of Internet of things, cloud computing, edge computing, and other technologies, like big data, machine learning [1, 2], the scale and organizational structure of the distributed system have been greatly changed. Today's distributed computing requires not only scalability but also

Regular research paper.

Q. Zuo

Information System Department, State Key Laboratory of NBC Protection for Civilian, Beijing, China

F. Wu (✉)

Department of Mathematics and Computer Science, West Virginia State University, Institute, WV, USA

e-mail: heng.wu@wvstateu.edu

F. Yan

Department of Information Technology, Central Research Institute of History and Documents, Beijing, China

S. Lu (✉)

College of Computer Science, Electronic Engineering, Hunan University, Changsha, China

e-mail: sftu@hnu.edu.cn

C.-d. Eduardo

Department of Computer Science, Midwestern State University, Wichita Falls, USA

e-mail: eduardo.colmenares-diaz@msutexas.edu

J. Liang

School of Computer and Electronics Information, Guangxi University, Guilin, China

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_65

909

unprecedented adaptability in dynamic scenarios. This requires that a flexible communication and interaction mechanism with dynamic and loose coupling characteristics should be adopted among the participants in the distributed system, so as to meet the requirements of large-scale decentralized control and dynamic change.

Publish/subscribe system is an intermediate system that enables information producers and information consumers to interact anonymously. It has many characteristics, such as loose coupling communication, dynamic plug and play, and asynchronous communication. At present, the most widely used methods are topic-based publish/subscribe and content-based publish/subscribe.

Compared with the theme-based publish/subscribe model, the content-based publish/subscribe model has a stronger ability to express information needs, so it can provide more accurate and efficient information sharing capabilities for distributed systems. Matching algorithms are one of the important research contents in a content-based publish/subscribe system. The efficiency of event matching not only determines the real-time performance of the system but also restricts the scalability of the whole system [3]. At present, there are many researches on the efficient and accurate matching algorithm [3–5], but they do not consider the fuzziness of user information demand and subscription. Reference [6] studies how to reduce the number of forwarding subscriptions by using the logical coverage relationship between subscription constraints in a fixed tree topology network, but this method is not suitable for dynamic wireless networks.

According to the characteristics of a dynamic wireless network, this paper studies the efficient fuzzy matching algorithm in content-based publish/subscribe. Based on the subjectivity and fuzziness of different users' understanding and expression of information needs, the fuzzy set theory is applied to information matching algorithms to improve the rationality of matching results. This paper also studies the methods to improve the efficiency of the fuzzy matching algorithm and reduce the maintenance cost of subscription information, proposes an efficient organization mode of subscription information, and verifies its efficiency through simulation experiments.

2 Content-Based Publish/Subscribe System

2.1 Introduction of Content-Based Publish/Subscribe System

The information distribution system based on the content-based publish/subscribe technology is a distributed system which makes the publishers and subscribers share information anonymously. It consists of the server and client. The server is a node of information distribution; each server provides services for a certain number of clients. As shown in Fig. 1. The system has the characteristics of asynchronous, loose coupling, and transparent transmission [8]. The information distribution mode

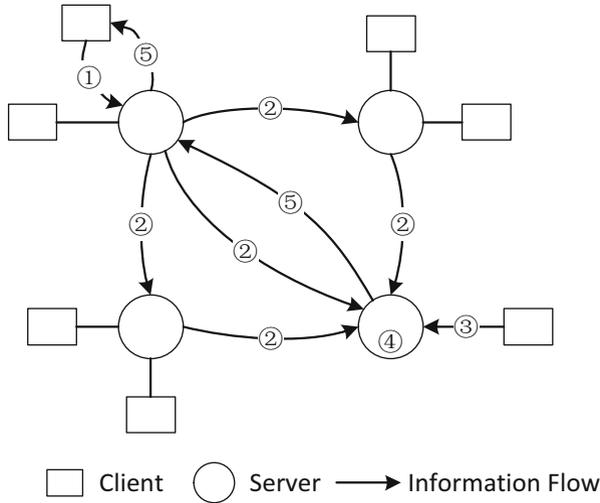


Fig. 1 Topology of the information distribution system

based on publish/subscribe is very suitable for the accurate distribution and sharing of information when the user’s demand is not clear.

The flow chart of information distribution is shown in Fig. 1. First, users subscribe to the server through the client according to the information requirements. In order to ensure that the published information can be accurately and timely distributed to the corresponding users, each server will synchronize the user’s subscription information in real time. Process ③ is a process in which users publish information (events) through publishing clients. Process ④ is the process that the server matches the user requirements with the events. Process ⑤ is the process of distributing to some users after they have successfully matched their subscriptions.

The topology of the wireless dynamic network will change dynamically with the change of network node location, so it is impossible to use the subscription routing method in reference [4] which uses the coverage relationship between subscription constraints to reduce the number of subscriptions and forwarding. In general, simple routing method can be used [8]. By broadcasting the added, changed, and unregistered subscription information in the distribution agent network, all nodes maintain the user’s subscription information synchronously, so the published events only need to match the user’s subscription at their local servers and then distribute to the matched subscribers. The simple routing method needs less network bandwidth, but the maintenance cost of subscription information is relatively high, so its organization mode and maintenance method will be more important. This paper will design an efficient organization mode of subscription information to solve this problem.

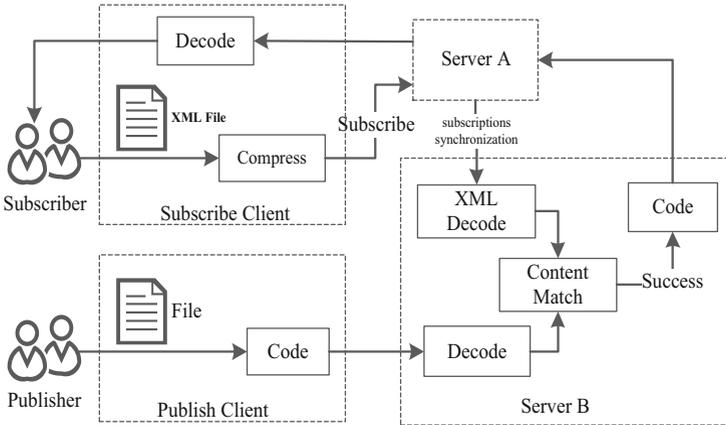


Fig. 2 Content-based distribution system model

2.2 Content-Based Publish/Subscribe Model

Before implementing the content-based publish/subscribe system, the event model and subscription model should be defined. The content of different kinds of information varies greatly. If a unified information description model is established, the event model will be too complex and contain too much redundant information, which is not conducive to improving the matching efficiency of events and subscriptions. Therefore, the current industry standard classification method can be used as the event model of the system. The content attributes that users can subscribe to a subset of a certain kind of information, which can be organized in XML language. The content-based information distribution system model is shown in Fig. 2.

3 Design of Efficient Fuzzy Matching Algorithm

At present, the content-based publish/subscribe system can be divided into two categories according to the different data organization methods of the event model: map-based and XML-based. In the map-based publish/subscribe system, the content of events is represented as a collection of “attribute = value.” The prototype system includes Gryphon, Siena, Jedi, Rebeca, etc. [9]. The information distribution system in this paper is essentially a map-based publish/subscribe system.

In the server shown in Fig. 2, the format message is decoded as a set of “attribute-value.” The XML file containing user subscription information is parsed into a set of multiple subscription constraints, $sub = \{C1, C2 Cn\}$, the subscription constraint

C_i is represented by a triple (Attribute, Predicate, Value), for example (height, \geq , 500 m) [9].

3.1 Concept and Process of Fuzzy Matching Algorithm

The existing publish/subscribe system adopts a precise matching algorithm, that is, an event either satisfies a subscription or does not satisfy the subscription. However, the information needs of users are subjective, and the expression of information needs is also inadequate, so the information needs expressed by users are fuzzy and inaccurate. To solve this problem, this paper designs a fuzzy matching algorithm which makes the matching more reasonable.

The basic idea of the algorithm is to treat the user's demand for information as a fuzzy set. If each information contains several subscribe attributes, the user's demand for information is a set of fuzzy sets. The distribution of each fuzzy set is determined according to the characteristics of content attributes and expert experience, and the parameters of membership function are determined by the actual subscription of users. The match between an event and a subscription is a match between multiple content properties in an event and multiple subscription constraints in a subscription. When some information is published, by calculating the membership degree of the event and corresponding fuzzy sets, we can determine whether the event matches the user's subscription successfully. In the precise matching algorithm, if an event property fails to match the subscription constraint, the matching between the whole event and the subscription will be judged as a failure. In the fuzzy matching algorithm, the matching results of some event content attributes and subscription constraints will not determine the final results, and comprehensive judgment is needed.

Generally, different content attributes have different importance to a user, so the membership degree of different content attributes has a different influence on the matching degree of the whole event and subscription in the fuzzy matching process. The more important the content attribute is to the user, the greater the role it plays in the matching of events and subscriptions. Therefore, the relative weight of each attribute can be determined by AHP first, and then the weighted average of membership degree of each content attribute can be calculated as the matching degree of the whole event and user subscription.

To sum up, the fuzzy matching process of a published information and a user's subscription information can be summarized as follows:

1. The membership function of each fuzzy set of subscription constraints in user subscription is determined. The membership function of subscription constraint C_i is $\mu_i(x)$.
2. Determine the weight of each content attribute in the battlefield information, and set the relative weight of attribute Attr _{i} in the event as ω_i .

3. The membership degree of each content attribute and the corresponding subscription constraint fuzzy set in the event are calculated. If the value of content attribute $Attr_i$ in the event is x_i , the membership degree of $Attr_i$ in the corresponding subscription constraint fuzzy set is $\mu_i(x_i)$.
4. Calculate the matching degree M between the publishing event and the user's subscription, M is gotten by Eq. 1.

$$M = [\mu_1(x_1) \cdots \mu_n(x_n)] \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix} = w_1\mu_1(x_1) + \cdots + w_n\mu_n(x_n) \quad (1)$$

5. Determine whether the publishing event matches the user's subscription successfully. If the matching degree M is greater than or equal to the matching degree threshold λ_M set by the user, the event matches the user's subscription successfully. Otherwise, the matching fails.

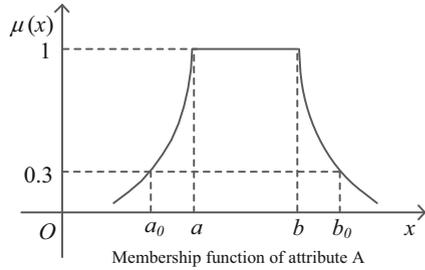
3.2 Efficient Fuzzy Matching Algorithm

The above method needs to calculate the matching degree between events and all users' subscriptions one by one and attribute by attribute; then it can finally determine which users' subscriptions match the events successfully. When the scale of the information distribution system and the number of users' subscriptions are large, the time efficiency of the fuzzy matching algorithm will be greatly reduced, so a method which can effectively improve the matching efficiency without reducing the rationality of fuzzy matching algorithm is needed.

One way to improve the matching efficiency is to narrow the subscription range of fuzzy matching. That is to say, through the existing precise matching algorithm, we can quickly select a part of users' subscriptions that are likely to match the publishing event successfully and only use the fuzzy matching algorithm to calculate the matching degree between this small part of subscription information and the publishing event.

In order to achieve the fast and accurate matching, the user's original subscription needs to be fuzzy preprocessed. After determining the membership function of the user's subscription constraint fuzzy set, a small membership value is taken as the membership fuzzy threshold value, and the subscription constraint value corresponding to the threshold value is taken as the new subscription constraint value of the user for accurate matching. As shown in Fig. 3, suppose that a certain information contains content attribute A, the membership function curve in the graph is determined by the subscription of a user, and the actual subscription of

Fig. 3 Membership function curve



a user to attribute a is interval $[a, b]$. Taking the membership degree of 0.3 as the fuzzy threshold, the subscription range of attribute A will be changed to the interval $[a_0, b_0]$ after fuzzy preprocessing. First of all, all the subscribe attributes of all users' subscription information are preprocessed by the abovementioned fuzzy method, and then the matching results can be obtained by fast and accurate matching with publishing events. The result is the set S_1 in Fig. 3.

It can be seen from the above analysis that all subscriptions need to match with publishing events exactly once, and then a part of users' subscriptions and publishing events can be filtered out for fuzzy matching. Therefore, the efficiency of the exact matching between events and subscriptions also restricts the efficiency of the whole fuzzy matching algorithm. The basic idea to improve the efficiency of accurate matching algorithm is to optimize the organizational structure of user subscription information and improve the time efficiency of matching by reducing the number of judgments on the same subscription constraints in the matching process [5].

3.3 Logical Coverage Relationship Between Subscription Constraints

Different users may subscribe to the same battlefield information through the same content attribute, so there will be a logical coverage relationship between subscription constraints in the subscription collection. For example, both user a and user b subscribe to the same information through the height attribute, where $C_{ai} = (\text{height} \geq 5000 \text{ m})$, $C_{bj} = (\text{height} \geq 10,000 \text{ m})$; then there is a logical coverage relationship between the subscription constraint C_{ai} and C_{bj} , that is, when the constraint C_{bj} is satisfied, the constraint C_{ai} must be satisfied.

Therefore, we can use the logical coverage relationship between constraints to reduce unnecessary comparison operations. For example, we can merge the same subscription constraints, arrange the constraint values in an array in order, and then use dichotomy to search. When the first subscription constraint value that matches the event property value successfully is found, the matching results of all subscription constraint values can be determined.

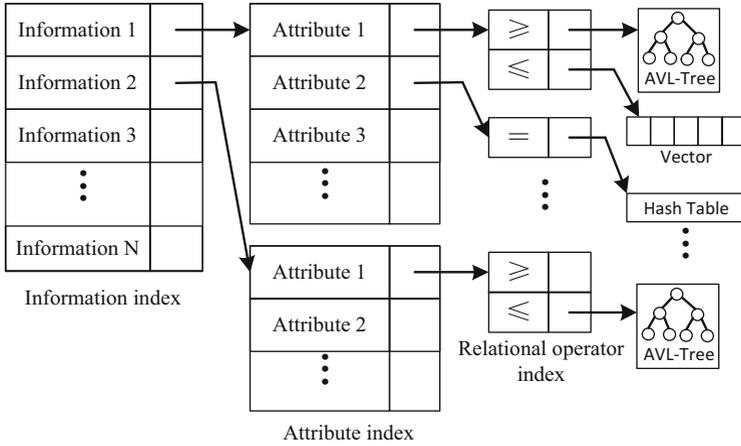


Fig. 4 Data structure of user subscription information

In order to use the logical coverage relationship between subscription constraints and improve the search efficiency of subscription information, a multilevel index structure is established as shown in Fig. 4.

In Fig. 4, the relational operator index points to the constraint value [10] of the information content attribute subscribed by the user. In order to organize a large number of subscription constraint values, the data structures that can be considered include sequential storage structure (such as an array, queue) and chain storage structure (such as linked list, binary tree).

3.4 Design of Subscription Information Organization Pattern in Matching Algorithm

In order to improve the time efficiency of content matching and reduce the maintenance cost of subscription information, it is necessary to design the organization mode of user subscription information in the information distribution system with a simple event routing method. Taking the sequential storage structure for storing user subscription information as an example, the constraint values of user subscription can be organized into three modes as shown in Fig. 5 by using the logical coverage relationship between constraints.

Mode (a) is a simple logic coverage mode. V_i is the constraint value arranged in ascending order. U_i is the user whose constraint value is V_i . When matching, binary search is carried out for V_i . After finding the constraint value v greater than or equal to the E_i , the matching results can be obtained by traversing the users behind v .

Mode (b) is an improvement on (a), which puts all users whose constraint value is greater than or equal to V_i into the queue. When matching, only binary search is

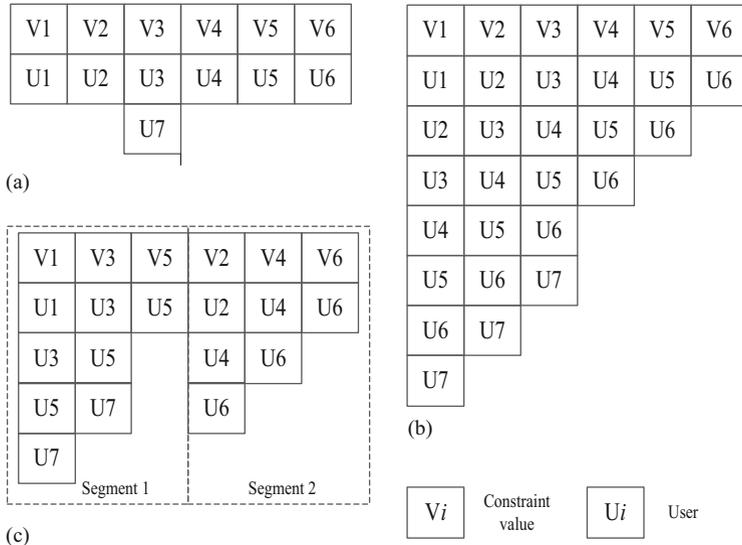


Fig. 5 Organization mode of constraint value: (a) simple logic coverage mode; (b) Complete logical coverage mode; (c) Segmentation full logical coverage mode

needed to get the matching results, which can effectively improve the time efficiency of matching. However, the space complexity of this mode is very poor, and the efficiency of subscription maintenance is also very low.

Mode (c) is the segment full logical coverage mode. In mode (c), the constraint values are not ordered, but they are ordered in the segment interval, and the complete logical coverage mode is adopted in each segment interval. When matching, binary search is carried out for each interval, and the final matching result is the combination of the matching results within each segment interval. In this mode, if the size of segment space is N , then the time complexity of matching is $O(\log N * n/N + n) = O(n)$, the space complexity is $O(1/2(N*(N + 1))*(n/N)) = O(n)$, and the time complexity of inserting subscription constraint value is $O(\log N + N) = \text{constant order}$.

4 Experiment and Analysis

In the experiment, the vector container is used to organize the constraint values of users. The subscription constraint value is simulated by the random number generated by the Box-Muller method, which conforms to the normal distribution. The distribution of the subscription constraint values of the three attributes is $N(500,100)$, $N(1000,300)$, and $N(1500,500)$. The mean value of the three normal distributions is taken as the published event, that is, $Event = \{500, 1000, 1500\}$.

Fig. 6 Comparison of matching efficiency in three modes

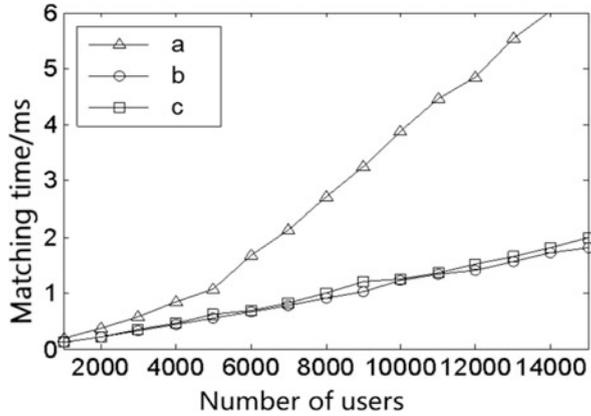
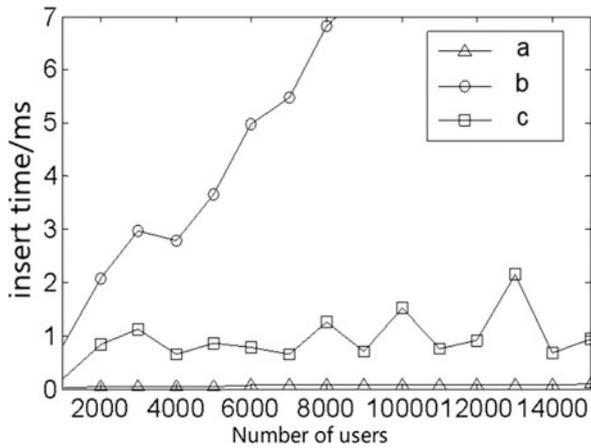


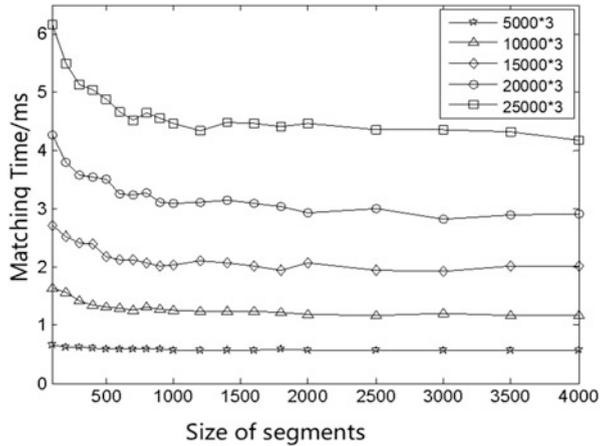
Fig. 7 Comparison of subscription maintenance efficiency in three modes



The experimental results are shown in Figs. 6, 7, and 8. The experimental results of the matching efficiency and subscription maintenance efficiency of the three modes are shown in Figs. 6 and 7, and the segmentation interval N of mode (c) is 1000. The experimental results show that mode (c) is an optimal scheme. Mode (c) can not only ensure high matching efficiency but also effectively control and reduce the cost of maintaining subscription information for distribution agent nodes.

Figure 8 shows the relationship between segment size N and matching time under different subscription data scales after adopting mode (c). It can be seen from the law of curve change that it is a negative exponential function. The optimal segment size is different under different user scales. In order to ensure the efficiency of the algorithm, the size of segment N should be increased with the increase of subscription data.

Fig. 8 The relationship between matching time and segment size



5 Conclusion

Firstly, this paper designs a content-based information distribution system model for the information sharing requirements of dynamic networks. Then, considering the subjectivity and fuzziness of users’ understanding and expression of information needs, an efficient fuzzy matching algorithm is designed based on the fuzzy set theory and the precise matching algorithm. Finally, in order to improve the time efficiency of the matching algorithm and reduce the maintenance cost of the subscription information, a segmented complete logical coverage pattern is designed to organize the subscription information, and the rationality of the pattern design is verified by experiments.

This algorithm can improve the rationality and efficiency of content matching in the information distribution system and is of great significance to the effective sharing of information. How to determine the membership function of information content attribute is the focus of future research.

References

1. S. Lu, Q. Zeng, H. Wu, A New Power Load Forecasting Model (SIndRNN): Independently Recurrent Neural Network Based on Softmax Kernel Function’, HPCC-2019
2. H. Wu, S. Lu, Temperature Prediction Based on Long Short Term Memory Networks, CSCSI’19
3. J. Chen, J. Shiguang, J. Pan, et al., Content-based fast event matching algorithm. *J. Commun.* **32**(6), 78–85 (2011)
4. L. Zeng, H. Yang, A trapezoidal matching algorithm in content-based publish and subscribe system. *Comput. Technol. Dev.* **22**(10), 1–4 (2012)
5. Y. Chen, J. Liu, A fast matching algorithm for content-based publish-subscribe system. *Microcomput. Appl.* **31**(2), 41–43 (2012)

6. Y. Pan, K. Zhang, J. Pan, Research on publish/order mechanism and algorithm based on predicate covering technology. *Comput. Res. Dev.* **48**(5), 765–777 (2011)
7. X. Hou, F. Gao, Summary of content-based publish and subscribe system. *Comput. Dev. Appl.* **27**(10), 10–13 (2014)
8. J. Ma, T. Huang, J. Wang, et al., Key technology of large-scale distributed computing publish and subscribe system. *J. Softw.* **1**(17), 134–147 (2006)
9. L. Dong, T. Gao, R. Qiu, Battlefield situation real-time distribution technology based on semantic publish and subscribe system. *Fire and Command Control* **42**(4), 110–113 (2017)
10. F. Fabret, H.A. Jacobsen, F. Llirbat, et al., *Filtering Algorithms and Implementation for Very Fast Publish/Subscribe Systems [M]* (ACM Press, New York, 2010)

Agile IT Service Management Frameworks and Standards: A Review



M. Mora, J. Marx-Gomez, F. Wang, and O. Diaz

1 Introduction

In the last two decades, several IT service management (ITSM) frameworks and standards to manage the planning, design, deployment, operation, and improvement of IT services have been used by business organizations [1]. The main ITSM frameworks and standards reported in the literature are ITIL v2011 [2], CMMI-SVC v1.3 [3] (Software Engineering Institute, 2010), and the ISO/IEC 20000 [4, 5].

The utilization of these ITSM frameworks and standards have produced relevant benefits to business organizations such as IT service quality improvement, IT service management cost reduction, and IT service user satisfaction increment [6]. However, their implementation demands also significant organizational resources (economic, human, and technological ones) and efforts (large implementation periods), which limits their successful utilization to very large-sized and large-sized business organizations [7]. For the case of medium-sized and small-sized organizations, the utilization of these ITSM frameworks and standards is not technically economically affordable [8–10]. Additionally, in the last decade, the

M. Mora (✉)

Information Systems, Autonomous University of Aguascalientes, Aguascalientes, Mexico
e-mail: jose.mora@edu.uaa.mx

J. Marx-Gomez

Informatics, University of Oldenburg, Oldenburg, Germany

F. Wang

Information Technology and Administration Management, Central Washington University, Ellensburg, WA, USA

O. Diaz

National Data Center, INEGI, Aguascalientes, Mexico

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_66

921

business environment has changed from stable and midterm user demands to dynamic and short-term ones, pushing business organizations to implement IT services from an agile perspective for supporting the digital disruption wave [11]. In the domain of software engineering [12], this agile perspective was proposed two decades ago [13], and it has strongly permeated most business organizations. Now, agile practices such as Scrum and XP are widely used [14]. In contrast, in the ITSM domain, the agile ITSM frameworks and standards have recently emerged [15, 16] and their empirical positive impacts and their implementation barriers are still unknown given the null or minimal empirical reported evidence in the literature. Consequently, ITSM practitioners and academics lack informative references on the applicability, benefits, and limitations of using agile ITSM frameworks and standards.

In this research, thus, we reviewed the main emergent ITSM frameworks and standards that are proffered as agile, from a conceptual-nonempirical-research approach. The four proffered agile ITSM frameworks and standards analyzed were ITIL v4 [17], VerisM [18], FitSM [19–21], and the ISO/IEC 20000–1:2018 [22]. The conceptual review approach was focused on the inclusion and adherence from these ITSM frameworks and standards to an agile aim, the agile values, the agile principles, and the agile practices proposed in two recent agile ITSM studies [15, 16]. Our research aim is to assess the extent of agility of these emergent proffered agile ITSM frameworks and standards regarding an agile ITSM scheme.

The remainder of this paper continues as follows. In Sect. 2, the background on ITSM, agile approach, and the agile ITSM scheme was reported. In Sect. 3, the review of the main four proffered agile ITSM frameworks and standards is presented. Finally, in Sect. 4, a discussion of the implications and conclusions of this research is reported.

2 Background on ITSM and Agile ITSM Tenets

This section reviews the background on ITSM and ITSM agile tenets.

2.1 *ITSM Background*

The IT management domain has adopted the service paradigm [23–25] from the industrial engineering [26] and the marketing [27] domains. IT service management (ITSM) was focused initially on IT operation processes (i.e., IT service support and IT service delivery processes) [23], but it evolved toward the full IT management area [24, 25]. Dedicated books on modern IT management topics [24, 25] account for the adoption of a service paradigm, as well as the emergence of specific IT service management frameworks such as ITIL v2011 [2], CMMI-SVC v1.3 [3], and ISO/IEC 20000 [4, 5].

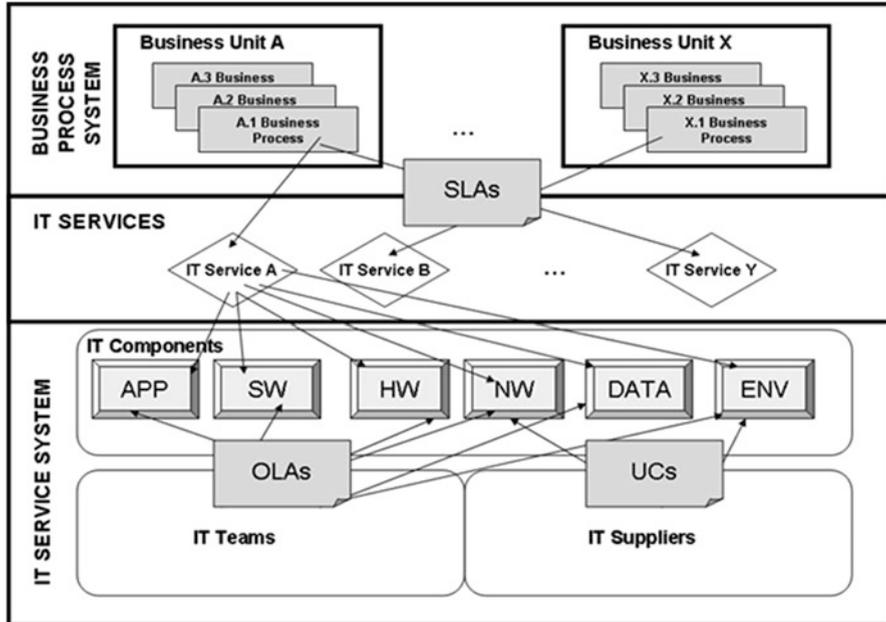


Fig. 1 The IT service system and IT service concept

Service management, in the ITSM domain, refers to the organizational capabilities used and applied to provide value to customers through the delivery of services [2]. Services are “means of delivering value to customers by facilitating outcomes customers want to achieve without the ownership of specific costs and risks” [2; p. 5]. ITSM is defined as “the implementation and management of quality IT services that meet the needs of the business” [2; p. 7].

An IT service is a service made up of IT, people, and processes; it is delivered by an IT service provider and consumed by an IT service customer. Both the IT service provider and customer form an IT service system. Thus, ITSM provides IT services “through an appropriate mix of people, process, and information technology” [2; p. 7]. Figure 1 portrays the concepts of IT services and the IT service system.

Value is realized in the IT service delivery when the IT service impacts on the utility (fit for purpose) and warranty (fit for use) on the customer business process supported by the IT service are achieved. The utility of an IT service corresponds to what the service does, its warranty, and how well it is delivered [2]. The utility of an IT service is achieved when occurs a performance improvement and/or a reduction of constraints on a customer’s business process using the IT service. Warranty of an IT service is received when the customer obtains the expected levels of availability, capacity, continuity, and security from the contracted IT service. Hence, ITSM can essentially be summarized as an IT service-centered management approach to provide value (i.e., utility and warranty) to IT service customers.

2.2 *Agile ITSM Background*

In the last decade, the high dynamism of business demands for IT services [11], as well as the growing utilization of agile practices in other domains (i.e., lean manufacturing [28] and agile software engineering [12–14]), has pushed the ITSM professional and academic communities to propose and elaborate agile versions of ITSM frameworks and standards [15, 16]. Four main ITSM frameworks and standards that claim to be agile or that can be assumed as agile are ITILv4 [17], VeriSM [18], FitSM [19–21], and the ISO/IEC 20000–1:2018 standard [22].

According to several studies on agile foundations [29, 30], agile practices are also considered lightweight ones, but not vice versa. Lightweight practices are shortened but still useful practices regarding the original heavy-oriented ones. Agile practices are also lightweight ones, but they need to be also flexible (i.e., to embrace changes), responsive (i.e., reactive to changes), rapid (i.e., applicable in relatively short periods), lean-seeking (i.e., simple, high-quality, and waste minimizing), and improvable (i.e., continually improved).

ITSM literature on the agile approach is still scarce, and consequently, ITSM practitioners lack informative references on the applicability, benefits, and limitations of using agile ITSM frameworks. Nevertheless, an ITSM literature review identified two studies on agile ITSM frameworks [15, 16] based on the Agile Manifesto from the software engineering domain [12]. Table 1 shows a summarized adaptation from the components (aim, values, and principles) of the proposal of the agile ITSM framework from [15] as it is reported in [16]. Table 1 also includes a list of the most used agile practices based on diverse current studies in the software engineering domain [14, 31–32], as well as the seven ones proposed in [15] for agile ITSM.

Hence, the two opposite approaches (i.e., rigor-oriented ITSM and the emergent agile-assumed ITSM) exhibit conflicts and tensions in their tenets (aim, values, principles, and practices) [16], and thus, a review of their tenets is worthy to guide practitioners and academics on their adequate utilization.

3 **Review of the Main Four Agile ITSM Frameworks and Standards**

The main four proffered agile ITSM frameworks and standards are individually described in this section, and each one is reviewed regarding their extent of adherence to the agile ITSM tenets.

Table 1 A basic agile ITSM framework

Tenet	Tenet description
Aim	Providing business value to its customers and users
Values	V1. Individuals and interactions over processes and tools V2. Working IT services over comprehensive documentation V3. Whole team collaboration over contracts V4. Responding changes over the following plans
Principles	<i>Outcome principles</i> P1. Customer satisfaction is the highest priority. P7. Customer value is <i>provided</i> as a primary measure of success. <i>Project principles</i> P2. Embrace changes. P3. Deliver frequently useful and warranted IT services. <i>Team principles</i> P4. Business and technical people work together daily. P5. Adequate work environment. P6. Face-to-face conversations within teams. P8. Keep sustainable work. P12. Break-times for reflection. P11. Self-organized team. <i>Design principles</i> P9. Simplicity with technical excellence. P10. Value simplicity—the art of maximizing the amount of work not necessary.
Practices	<i>Agile ITSM practices</i> Pr.1A Self-Organized Teams. Pr.1B Coaching. Pr.2A Work Monitoring. Pr.2B Team Decision-Making. Pr.3A Focus on User Value. Pr.3B User concerns in SLA. Pr.4A Business Alignment. Pr.5A Work Integrated Teams. Pr.5B Work Face-to-Face Coordination. Pr.6A Simple Knowledge Management System. Pr.7A Operational and Project Dual Roles. <i>Agile software engineering practices</i> Pr.1 Daily Stand-Up Meetings. Pr.2 Sprint Planning. Pr.3 Sprint/Iteration. Pr.4 Short-Releases. Pr.5 Retrospectives. Pr.6 Face-to-Face Communication. Pr.7 Unit Testing. Pr.8 Tracking Monitoring. Pr.9 Continuous Integration. Pr.10 User Stories / Backlog. Pr.11 Team Working. Pr.12 Sprint Review. Pr.13 Coding Standards. Pr.14 Refactoring. Pr.15 Collective Ownership. Pr.16 40-hour per week. Pr.17 Simple Incremental Design. Pr.18 Simple Documentation. Pr.19 Burn-Down Charts. Pr.20 Release Planning Game. Pr.21 Backlog Grooming. Pr.22 Agile Dev-Test Team. Pr. 23 Acceptance Testing. Pr.24 Scrum of Scrums. Pr.25 Kanban. Pr.26 Dedicated Customer/Product Owner. Pr.27 Short Releases. Pr.28 Test-Driven Development. Pr.29 One Team Office. Pr.30 Agile UX. Pr.31 Scrum Master. Pr.32 Niko-Niko Calendar

3.1 ITIL v4

ITIL v4 [17; p. 14] has been redefined and restructured as a service value system (SVS) “to ensure a flexible, coordinated, and integrated system for the effective governance and management of IT-enabled services.” ITIL v4 does not claim to be an SVS for the whole organization, but it indicates that with the new business dynamic demands for digital transformations, enhanced customer experiences based on IT, and the proliferation of new practices and technologies such as agile paradigm, DevOps, Lean, cloud computing, Internet of Things, and machine

learning, the majority of the business services are IT-based enabled services. Thus, an updated ITSM framework is required.

ITIL v4 keeps the concept of service management as “a set of specialized organizational capabilities for enabling value for customers in the form of services” [17; p. 18]. However, the focus on the five-phase IT service lifecycle model (i.e., service strategy, design, transition, operation, and continual improvement) has been restructured in the concept of the six-phase service value chain, which is one of the five core components of the new SVS. However, these five IT service lifecycle phases have been implicitly included and updated in the six-phase service value chain. The concept of service is kept as “a means of enabling value co-creation by facilitating outcomes that customers want to achieve, without the customer having to manage specific costs and risks” [17; p. 248], and the concept of IT service is simplified to “a service based on the use of information technology” [17; p. 242]. The concept of value which was indirectly defined as how well an IT service helps to achieve the expected customer’s outcomes for using such an IT service now has been explicitly defined as “the perceived benefits, usefulness, and importance of something” [17; p. 20]. In particular, the concept of ITSM is not explicitly defined in ITIL v4.

The five core components of the ITIL v4 SVS are ITIL v4 service value chain (ITIL v4 SVC), ITIL v4 practices, ITIL v4 guiding principles, governance, and continual improvement. The six-phase ITIL v4 SVC defines a flexible and adaptable operational model for creating, delivering, and continually improving IT services. The ITIL v4 principles aim to guide organizational decision-making and behaviors toward an adequate service management culture to be applied from top to bottom organizational levels. There are eight principles. The ITIL v4 practices are organizational resources which guide it on what work to do. There are three categories of ITIL v4 practices (general management, service management, and technical management). ITIL v4 governance refers to the top-level policy and regulation body created to assure the alignment of the IT actions with the IT strategies, policies, and regulations. The ITIL v4 continual improvement model is reported as usable and required for all organizational areas from strategic to operational levels. The previous seven-phase model from ITIL v3-v2011 is supported.

The four ITIL v4 dimensions represent viewpoints on the ITIL v4 SVS, and these are fundamental for achieving effective and efficient service management that delivers IT services and/or IT products with the expected value. These dimensions are organizations and people, information and technology, partners and suppliers, and value streams and processes. Important is the consideration of political, economic, social, technological, legal, and environmental factors which by their external nature are out of the control of the ITIL v4 SVS but which must be considered because they constraint and influence the four ITIL v4 dimensions.

In the updated ITIL v4 ITSM framework, there are not mandatory or suggested obligatory IT practices to be performed. The new flexible and adaptable ITIL v4 SVS model, like the VeriSM framework, defines a generic six-phase SVC (plan, improve, engage, design and transition, obtain/build, and deliver and support). This six-phase SVC can accommodate flexibly the utilization of the 34 ITIL v4

practices (of which 14 is general management, 17 service management, and 3 technical management). These 34 ITIL v4 practices are not restricted to be used in a specific phase of the ITIL v4 SVC. Instead, there is a heat map reported for each practice to show where it is expected to use (but not in mandatory status) such a practice. Consequently, ITIL v4 proposes a flexible, adaptable, and highly customized service management model where each organization is responsible to define its value streams. Value streams are “specific combinations of activities and practices, and each one is designed for a particular scenario” [17; p. 83]. A value stream starts with the customer’s demand and ends with the delivery of value to such a customer. Value streams can be organized as disciplined, agile, or hybrid flexible workflows, and it is an organizational decision. Furthermore, some value streams, for their criticality level, can be designed for a disciplined approach and others with more flexible approaches (i.e., Agile, Lean, DevOps).

3.2 *VeriSM*

VeriSM [18; p. 376] is defined as a “value-driven, evolving, responsive, and integrated service management approach” for the entire organization in the digital era, and not only for the IT area. A service management approach is a management approach to deliver value to customers through quality products and services. VeriSM indicates that whereas the ITSM best practices frameworks have provided value to organizations in the last decade, the new digital business era demands a broader IT-based or digital transformation approach for the entire organization, and thus, these ITSM frameworks are insufficient to cope with the business demands in this digital era. VeriSM aims to help organizations on how they can use integrally a mesh of best management practices in a flexible way to deliver the right product or service at the right time to their customers. VeriSM is documented with a service management operating model composed of consumers, governance, service management principles, and the management mesh. The implementation of the VeriSM approach enables organizations to define governance requirements, service management principles, a management mesh of best practices, and the service or product stages from the definition, production, responding, and provision.

Customers provide the product or service requirements, pay, receive, and give feedback for the products or services. Governance provides the background system to direct and regulate the activities of an organization, and management provides the foreground system which manages the activities of an organization into the boundaries and regulations fixed by governance. Governance consists of three main activities (evaluate, direct, and monitor). Evaluate refers to compare the overall current organizational status vs the future forecasted or planned ones. Direct refers to create organizational principles, policies, and strategies. Monitor refers to assure that policies comply, and strategic performance are the expected ones. Service management principles are statements that define how the organization wants to perform and what is valued. Service management principles, thus, help to define

the specific best practices to include in the management mesh. Service management principles address usually assets/resources utilization, change, continuity, financial, knowledge, measurement and reporting, performance, quality, regulations, risk, and security issues.

Management mesh refers to the integral and flexible fabric composed of organizational resources, management practices, current and emergent technologies, and environmental conditions. This management mesh enables a flexible and agile management service approach in organizations to define, produce, provide, and respond to their products and services. The definition of a particular management mesh happens after the definition of governance strategies and policies, and service management principles. In particular, the environmental conditions in the management mesh include the called service stabilizers (processes, tools, and measurements). This management mesh lately defines four functional areas/stages for developing and providing the products and services of the organization. These are define, produce, provide, and respond. There are four, three, three, and two high-level activities, respectively, in the four stages. The four ones of the Define stage are consumer need, required outcome, solution, and service blueprint. The three ones of the Produce stage build, test, and implement and validate. The three ones of Provide are protect, measure, and maintain and improve. The two ones of Respond are record and manage.

3.3 *FitSM*

FitSM [19] is defined as “a lightweight standards family” in the ITSM domain compatible with the ISO/IEC 20000 standard and ITIL v3-v2011 ITSM framework. FitSM aims “to maintain a clear, pragmatic, lightweight, and achievable standard that allows for effective IT service management (ITSM)” [19; p. 1]. FitSM is composed of seven documents. FitSM-0 overview and vocabulary, FitSM-1 requirements, FitSM-2 objectives and activities, FitSM-3 role model, FitSM-4 selected templates and samples, FitSM-5 selected implementation guides, and FitSM-6 maturity and capability assessment scheme. FitSM claims its application in any type of organization and IT area.

In FitSM-1, reported are seven categories of general requirements for a service management system which include 16 items, as well as the 14 FitSM processes with their 69 specific requirements. The seven categories of general requirements are top management commitment and responsibility, documentation, defining the scope of service management, planning service management, implementing service management, monitoring and reviewing service management, and continually improving service management.

The 14 processes of FitSM are Service Portfolio Management (SPM), Service Level Management (SLM), Service Reporting Management (SRM), Service Availability and Continuity Management (SACM), Capacity Management (CAPM), Information Security Management (ISM), Customer Relationship Man-

agement (CRM), Supplier Relationship Management (SUPPM), Incident and Service Request Management (ISRM), Problem management (PM), Configuration Management (CONFM), Change Management (CHM), Release and Deployment Management (RDM), and Continual Service Improvement Management (CSI).

These 14 processes of FitSM are claimed to comply with the ISO/IEC 20001–1 standard [20]. Each process in FitSM is structured with an objective, setup activities, inputs, ongoing activities, and outputs. Additionally, there are seven objectives for the respective seven categories of general requirements for a service management system. FitSM defines also seven generic roles and about three one-specific-process roles for each one of the 14 FitSM processes. The seven generic roles are SMS owner, SMS manager, service owner, process owner, process manager, case owner, and member of process staff (process practitioner). FitSM reports in an ITSM documentation checklist guide over 60 artifacts.

FitSM claims to be a lightweight ITSM framework with a reduction to four core documents with a total of 38 pages compared with the extensive official documentation of full ITSM frameworks such as ITIL v3-v2011 and the ISO/IEC 20000. The concept of agile is not explicitly reported in the four core documents.

3.4 ISO/IEC 20000-1:2018

The ISO/IEC 20000 standard [22] has been reviewed for the third time from their first 2005 and second 2011 versions. This standard establishes the requirements for any organization (any type, any size) that can devise, implement, operate, and improve a service management system (SMS).

A management system is defined by the ISO/IEC 20000 standard [22; p. 3] as a system of “interacting elements of an organization to establish policies and objectives and processes to achieve those objectives”. Service management, in turn, is defined as a “set of capabilities and processes to direct and control the organization’s activities and resources for the planning, design, transition, delivery, and improvement of services to deliver value” [22; p. 9]. Consequently, an SMS refers to a management system for directing and controlling the organization’s service management activities. For the ISO/IEC 20000 standard [22], an SMS is focused on supporting the service lifecycle. A service is defined as a “means of delivering value for the customer by facilitating outcomes the customer wants to achieve” [22; p. 8]. Value refers directly to the extent of importance, benefit, or usefulness, assigned by the service customers/users.

The ISO/IEC 20000 standard (2018) is organized in a set of seven clauses. These are the context of the organization, leadership, planning, support of the SMS, operation of the SMS, performance evaluation, and improvement. Organizations interested in conforming to this standard are free on how to implement these seven categories of clauses but are also obligated to implement all of them. The ISO/IEC 20000 standard establishes [22; p. vii] that “an SMS as designed by an organization, cannot exclude any of the requirements specified in this document.”

The context of the organization clause contains four subclauses. They refer to the understanding organization and its context, understanding of needs and expectations from stakeholders, determining the scope of the SMS, and devising, implementing, operating, and improving the SMS. The leadership clause contains three subclauses. They refer to governance issues such as leadership and commitment, policy, and organizational roles, responsibilities, and authorities. The planning clause contains three subclauses. They refer to risks and opportunities, objectives, and plan the SMS. The support of the SMS clause contains six subclauses. They refer to resources, competence, awareness, communication, documented information, and knowledge. The operation of the SMS clause contains six subclauses. They refer to operation, planning, and control, service portfolio, relationship and agreement, supply and demand, service design, build and transition, resolution, and fulfillment, and service assurance. The performance and evaluation clause contains four subclauses. They refer to monitoring, measurement, analysis, and evaluation, internal audit, management review, and service reporting. Finally, the improvement clause contains two subclauses. These are nonconformity and correction actions, and continual improvement.

According to the ISO/IEC 20000 standard [22; p. vii], it can be used in combination with most accepted ITSM frameworks (i.e., ITIL v2011, and CMMI-SVC v1.3). This standard was released in 2018, before the release of ITIL v4, and consequently is aligned more to the rigor-oriented ITSM approach than the agile one. The core category of clauses of this ISO/IEC 20000 standard corresponds to the operation of the SMS, performance evaluation, and improvement. Operation of the SMS category defines specific requirements for establishing performance criteria and controlling mechanisms for the SMS processes; enacting a service delivery process based on a service portfolio of planned, underdevelopment, active, and removed services; planning services; controlling the involved parties in the service lifecycle; managing the service catalogue; managing assets; managing configurations; managing business relationships with customers and users; managing service levels; managing external suppliers; managing internal suppliers; managing budgets and accounting services; managing service demand; managing service capacity; managing service changes; planning, designing, building, transitioning, deploying, and releasing services; managing incidents; managing service requests; managing problems; managing service availability; managing service continuity; managing information security; monitoring, measuring, analyzing, and evaluating services; internal auditing; management review; service reporting; nonconformity and corrective actions; and continual improvement.

Hence, the ISO/IEC 20000 standard [22], being aligned to the ITIL v2011 framework, despite its claimed update, presents still a heavy-process oriented approach.

3.5 Analysis of the Proffered Agile ITSM Frameworks and Standards

Table 2 reports the evaluation for four claimed agile ITSM frameworks and standards. As it was indicated, ITIL v4 and FitSM are focused on the IT area, while VeriSM and the ISO/IEC 20000–1:2018 claim to be a whole organizational service management approach, but this also applies to the IT area. The ordinal scale used for this conceptual evaluation of each attribute was as follows: (1) weak: agile statements are weakly supported; (2) moderate: agile statements are supported but not explicitly included, and (3) strong: agile statements are supported and explicitly included. An overall evaluation was also conducted based on the individual evaluations for each attribute. The scale use was as follows: (1) weak, when the majority of the individual evaluations were weak; (2) weak-moderate, when there was a mixed of weak and moderate individual evaluations; (3) moderate, when the majority of the individual evaluations were moderate; (4) moderate-strong, when there was a mixed of moderate and strong individual evaluations; and (5) strong, when the majority of the individual evaluations were strong.

4 Discussion of Implications and Conclusions

In this section, a discussion of theoretical and practical implications, as well as the conclusions and recommendations for further research are reported.

4.1 Discussion of Implications

The results from Table 2 on the adherence to agile ITSM tenets for the four proffered ITSM frameworks and standards indicate that ITIL v4 and VeriSM can be considered with moderate-strong and strong adherence to agile tenets. FitSM was assessed with a weak-moderate level and the ISO/IEC 20000–1:2018 standard with a weak level. These results are also congruent with their reasons to be proposed.

ITIL v4 emerged with the consideration of a new business dynamic that demands the inclusion of several digital practices and technologies such as agile paradigm, DevOps, Lean, cloud computing, Internet of Things, and machine learning. ITIL v4 considers that the majority of the business services are supported by IT-based enabled services, and thus, an updated ITSM framework was required. ITIL v4 was restructured from the five-phase service lifecycle to a six-phase service value chain (SVC), and it defines a flexible and adaptable operational model for creating, delivering, and continually improving IT services. In this updated ITIL v4 ITSM framework, there are not mandatory IT practices to be performed. This six-phase SVC can accommodate flexibly the utilization of the 34 ITIL v4 practices, and they

Table 2 A basic agile ITSM framework

Tenet	Tenet description	ITIL v4	FitSM	VeriSM	ISO/IEC 20000-1:2018
Aim	Providing business value to its customers and users	Strong	Weak	Strong	Strong
Values	V1. Individuals and interactions over processes and tools. V2. Working IT services over comprehensive documentation. V3. Whole team collaboration over contracts. V4. Responding to changes over the following plans.	Strong	Moderate	Strong	Weak
Principles	<i>Outcome principles</i> <i>Project principles</i> <i>Team principles</i> <i>Design principles</i>	Strong Strong Moderate Strong	Weak Moderate Weak Moderate	Strong Moderate Strong Strong	Strong Weak Weak Moderate
Practices	<i>Agile ITSM practices</i> <i>Agile software engineering practices</i>	Moderate Moderate	Moderate Weak	Strong Moderate	Weak Weak
Overall agility Evaluation		<i>Moderate-strong</i>	<i>Weak-moderate</i>	<i>Strong</i>	<i>Weak</i>

are not restricted to be used in a specific phase of the ITIL v4 SVC. Their utilization is rather suggested through a visual heat map reported for each practice to show where it is expected to be used. ITIL v4 framework was assessed as moderate to strong agility level because it was identified that some agile tenets are not covered in its core structure, despite some of them are reported as complementary practices to be used jointly with ITIL v4 practices. However, this assessment moderate-strong qualifies rather as an agile than a rigorous ITSM framework.

VeriSM has also emerged to cope with the dynamic demands caused by the digital transformation era, as well as by the varied availability of management approaches, frameworks, practices, and methodologies, so its concept of management mesh, to elaborate an ITSM fabric customized. VeriSM, thus, does not impose mandatory low-level activities to be followed (but recommended), but its four high-level phase model on how services or products are developed is expected to be followed. VeriSM's official documentation includes emergent ITSM management practices such as Lean, DevOps, and customer/user experience, as well as emergent technologies such as cloud computing, machine learning, and the Internet of Things. Thus, VeriSM can be called an "open-mind alike" service management approach which can glue all the management approaches and emergent technologies.

The assessment for FitSM was from weak to moderate. Some agile tenets (aim, outcome principles, team principles, and agile SwE practices) are weak and the remaining ones are moderate. The four official FitSM documents do not report the concept of agility. FitSM emerged in the context of scientific data centers (cloud, grid, and federation types), providing scientific computing services to a wide global community. Consequently, while an ITSM process framework was required, the available ones (ITIL v2011 and ISO/IEC 20000) were considered quite bureaucratic with excessive required documentation, most likely useful for business organizations. FitSM, thus, emerged with the need to lighten this heavy-process approach rather than provide an agile one.

The ISO/IEC 20000-1:2018 standard was included in this review because it was recently updated as a third version. Considering that it was released under the new highly dynamic business environment with strong demands for agile delivery of IT business digital services, it was expected that this standard would present an agile view. However, it was identified that this standard kept its heavy-process oriented approach, and thus, its agility assessment was weak. Individually, the aim and outcome principles of tenets were evaluated as strong, but the remainder agile were evaluated as weak.

5 Conclusions

This research reviewed the main four agile proffered ITSM frameworks (ITIL v4, VeriSM, and FitSM) and standards (ISO/IEC 20000-1:2018) reported in the current ITSM literature, to assess their coverage to agile tenets. This assessment is worthy given the current growing interest and needs to implement successful agile ITSM

approaches due to the new business environment driven by digital transformation pressures.

It was identified that two of the four ITSM frameworks (VeriSM and ITIL v4) can be considered as agile ITSM ones. VeriSM was assessed as strong agile and ITIL v4 as a moderate-strong agile one. In contrast, the ITSM framework FitSM and the ISO/IEC 20000-1:2018 standard were expected to be also evaluated as moderate or strong agile, but they qualified as weak-moderate and weak agile, respectively.

These two ITSM, framework and standard, can be considered rather lightweight but not agile ones. VeriSM, as a generic service management approach that can be used also for the IT area, presented an adequate flexible approach that can accommodate effortlessly an agile approach, and thus, it was assessed as strong in its adherence to agile tenets. ITIL v4 was restructured also for fitting agile practices, and except for some missed agile tenets, this ITSM framework provides also a flexible and customizable ITSM framework of practices.

This review of the agile proffered ITSM frameworks and standards suggests the following statements: (1) clear agile ITSM frameworks will be demanded by the business organizations; (2) an adequate agile version of FitSM can be elaborated; (3) detailed implementation guides on how VeriSM and ITIL v4 can be applied are required; (4) VeriSM will expand their application in multiple global organizations; (5) specific agile version of the ISO/IEC 20000 standard can be generated; and (6) a globally accepted agile ITSM manifesto and framework with specific agile tenets alike the existing one in the software engineering field since two decades can be elaborated.

This review was conducted using the official documents from the four ITSM frameworks and standards by the first two authors and reviewed by the third one. The fourth author reviewed the logical consistency of this study from an ITSM practitioner perspective. Consequently, there is a methodological limitation on the qualitative interpretations assessed.

Finally, it can be concluded that ITSM practitioners and academics can count on two agile ITSM frameworks at present (VeriSM and ITIL v4), but their adequate utilization and impacts must be further researched.

References

1. J. Iden, T.R. Eikebrokk, Implementing IT service management: A systematic literature review. *Intern. J. Inf. Manag.* **33**(3), 512–523 (2013)
2. itSMF UK, *ITIL Foundation Handbook* (The Stationery Office, London, 2012)
3. SEI, *CMMI for Service Version 1.3, CMU/SEI-2010-TR-034* (Software Engineering Institute, Pittsburgh, 2010)
4. ISO/IEC, *ISO/IEC 20000-1:2005 Information technology – Service Management – Part 1: Specification* (International Organization for Standardization, Geneva, 2005)
5. ISO/IEC, *ISO/IEC 20000-1:2005 Information technology – Service Management – Part 2: Code of Practice* (International Organization for Standardization, Geneva, 2005)

6. M. Marrone, F. Gacenga, A. Cater-Steel, L. Kolbe, IT service management: A cross-national study of ITIL adoption. *Commun.Assoc. Inf. Syst.* **34**(1), 49 (2014)
7. T.R. Eikebrokk, J. Iden, Strategising IT service management through ITIL implementation: model and empirical test. *Total Qual. Manag. Bus. Excell.* **28**(3–4), 238–265 (2017)
8. P. Küller, M. Vogt, D. Hertweck, M. Grabowski, IT service management for small and medium-sized enterprises: a domain specific approach. *J. Innov. Manag. Small Medium Enterp.* **1**, 1–17 (2012)
9. M. Ciesielska, implementation of service management system in small businesses: problems and success factors. *CER Comp. Eur. Res.* **2014**, 22–26 (2014)
10. K. Melendez, A. Dávila, M. Pessoa, Information technology service management models applied to medium and small organizations: A systematic literature review. *Comp. Stand. Interfaces* **47**, 120–127 (2016)
11. D.A. Skog, H. Wimelius, J. Sandberg, Digital disruption. *Bus. Inform. Syst. Eng.* **60**(5), 431–437 (2018)
12. R. Hoda, N. Salleh, J. Grundy, The rise and evolution of agile software development. *IEEE Softw.* **35**(5), 58–63 (2018)
13. J. Highsmith, A. Cockburn, Agile software development: The business of innovation. *Computer* **34**(9), 120–127 (2001)
14. VersionOne. CollabNet 13th Annual State of Agile Report (2018). Available from <https://www.stateofagile.com/#ufh-i-521251909-13th-annual-state-of-agile-report/473508>
15. B. Verlaine, Toward an agile IT service management framework. *Serv. Sci.* **9**(4), 263–274 (2017)
16. M. Mora, F. Wang, J.M. Gómez, O. Díaz, *A Comparative Review on the Agile Tenets in the IT Service Management and the Software Engineering Domains. In International Conference on Software Process Improvement* (Springer, Cham, 2019), pp. 102–115
17. TSO, *ITIL Foundation: ITIL*, 4th edn. (The Stationery Office Ltd., London, 2019)
18. C. Agutter, S. van Hove, R. Steinberg, R. England, *VeriSM – A Service Management Approach for the Digital Age* (Van Haren, Zaltbommel, 2017)
19. FitSM. FitSM-1: Requirements, The FitSM Standard Family: Standard for lightweight IT service management 1, version 2.1 (2016). Available from <https://www.fitsm.eu/downloads/>
20. FitSM. FitSM-2: Objectives and Activities, The FitSM Standard Family: Standard for lightweight IT service management 1, version 2.2 (2016). Available from <https://www.fitsm.eu/downloads/>
21. FitSM. FitSM-3: Role Model, The FitSM Standard Family: Standard for lightweight IT service management, version 2.2 (2016). Available from <https://www.fitsm.eu/downloads/>
22. ISO/IEC, *ISO/IEC 20000-1:2018 Information technology — Service Management — Part 1: Service Management System Requirements* (International Standards Organizations, Geneva, 2018)
23. S.D. Galup, R. Dattero, J.J. Quan, S. Conger, An overview of IT service management. *Commun. ACM* **52**(5), 124–127 (2009)
24. A.J. Keel, M.A. Orr, R.R. Hernandez, E.A. Patrocínio, J. Bouchard, From a technology-oriented to a service-oriented approach to IT management. *IBM Syst. J.* **46**(3), 549–564 (2007)
25. L. Pilorget, T. Schell, *IT Management: The Art of Managing IT Based on a Solid Framework Leveraging the Company's Political Ecosystem* (Springer, Wiesbaden, 2018)
26. R.B. Chase, U.M. Apte, A history of research in service operations: What's the big idea? *J. Oper. Manag.* **25**(2), 375–386 (2007)
27. S.L. Vargo, R.F. Lusch, The four service marketing myths: remnants of a goods-based, manufacturing model. *J. Serv. Res.* **6**(4), 324–335 (2004)
28. R. Shah, P.T. Ward, Lean manufacturing: context, practice bundles, and performance. *J. Oper. Manag.* **21**(2), 129–149 (2003)
29. K. Conboy, B. Fitzgerald, Toward a conceptual framework of agile methods, in *Extreme Programming and Agile Methods – XP/Agile Universe 2004, LNCS*, ed. by C. Zannier, H. Erdogmus, L. Lindstrom, vol. 3134, (Springer, Berlin, 2004), pp. 105–116

30. A. Qumer, B. Henderson-Sellers, An evaluation of the degree of agility in six agile methods and its applicability for method engineering. *Inform. Software Technol.* **50**(4), 280–295 (2008)
31. N. Kurapati, V.S.C. Manyam, K. Petersen, Agile software development practice adoption survey, in *Agile Processes in Software Engineering and Extreme Programming, XP 2012, LNBIP*, ed. by C. Wohlin, vol. 111, (Springer, Berlin, 2012), pp. 16–30
32. H. Alahyari, R.B. Svensson, T. Gorschek, A study of value in agile software development organizations. *J. Syst. Softw.* **125**, 271–288 (2017)

Contingency Planning: Prioritizing Your Resources



Kathryne Burton, Necole Cuffee, Darius Neclos, Samuel Olatunbosun, and Taiwo Ajani

1 Background

A contingency plan can be described as a proactive and comprehensive backup plan for any business. It is activated in the event of any type of a situational disaster, including natural, technological, or manmade, that may disturb employees, machines, or IT systems. A contingency plan may consist of rerouting data, emergency generators for power, escape routes for employees, and supervisory duties for contingency team members. Plans to get production up and running despite unforeseen circumstances can be the difference between a company that survives a disaster and one that folds [1]. There may be a cost associated with devising a contingency plan and maintaining it, but it could be minimal when measured against the cost of production loss [2].

Developing a well-rounded contingency plan includes analyzing all risks, first. This includes listing all possible events that could disrupt operations [1]. Next, a business should determine the likelihood and impact of all risks and prioritize them. A “risk probability chart” is a resource used to help evaluate and prioritize risks based on the severity of their impact and the probability of the event occurring. Next, businesses must create each event. Creating separate plans will outline the actions that should be taken if the risk occurs. Businesses must consider what must be done in order to resume normal operations after the impact of the event.

K. Burton · N. Cuffee · D. Neclos · S. Olatunbosun (✉)
Department of Computer Science, Norfolk State University, Virginia, VA, USA
e-mail: k.a.burton102699@spartans.nsu.edu; n.a.cuffee@spartans.nsu.edu;
d.j.neclos@spartans.nsu.edu; sbolatunbosun@nsu.edu

T. Ajani
Department of Computer Information Systems, Ferrum College, Virginia, VA, USA
e-mail: tajani@ferrum.edu

During this step, businesses should also clarify employee responsibilities, timelines that highlight when things should be done and completed after the event, restoring and communications processes, and the steps needed to take in advance to prevent losses when the event has taken place. Lastly, businesses should share, maintain, and execute the plan if necessary. Once the plan is completed, it should be quickly accessible to all employees and stakeholders.

In all, the best contingency plans benefit the company during a disaster. In most cases, a contingency plan helps minimize the loss of production. If implemented correctly, such plans show employees exact roles and responsibilities, while maximizing time and allowing the focus to be solely on the issue at hand. Most importantly, it creates the space to feel more prepared.

2 Literature Review

Technology is only as durable and reliable as it's created to be. It is not exempted from flaws. Indeed, It has a dual nature. It comes with risks in terms of adverse events and potential losses that can be due to several factors and may lead to the disruption of business operations [3]. Examples could be natural disasters, zero-day attacks, outages, etc.

An important factor for contingency planning is to construct that plan as a thorough guideline. According to Yiwen Shi, a reliable contingency plan should be standardized such that each stage adopts the consequence attained from the last stage and extends to a more detailed production [4]. This includes components such as business impact analysis, disaster recovery, and business continuity planning.

There are several methods in creating a contingency plan. No matter the method, the key to a successful contingency plan is to have a consistent flow between stages. A common method used is a rational unified process (RUP). A rational unified process is considered to be a method of iterative development or, in simpler terms, a waterfall methodology [4]. An RUP consists of seven stages, and though each stage is dependent on the next stage, there is space and flexibility to revert back to a previous stage in case new urgent scenarios appear.

Contingency planning should always be viewed as an immediate response procedure. When creating the contingency plan, refrain from attaching resources that will not be immediately available, and in addition, it is best practice to maintain practicality throughout the entire plan to ensure an immediate response [5]. It is important to note that risk assessments and contingency plans coincide with each other. Editor Joseph A. Schafer expresses that contingency planning should only focus on "foreseeable risk [5]." This statement is very agreeable. In today's society, whether it is the provided service or not, technology is the main driver for most companies. No matter the company's complexity, there will be several risks within the company's environment. Creating contingency plans for every risk is not only time-consuming but also costly.

There should only be one main objective when developing a contingency plan, and that involves the ability to continue business operations in the event of any major conflicting situations. Contingency planning should answer many questions, one in particular, “What is the likelihood that this contingency plan will be used?” Most companies create multiple plans in reference to their company’s risks and policies; however, as stated before, every risk does not need a contingency plan. With that in mind, as an information technology and/or information security professional, it is best practice to warrant purpose and necessity within the plan. If the contingency plan is purposeful and holds necessity, the likelihood of using it will be greater and more efficient for the company.

3 Contingency Planning

3.1 Research Design

Research was conducted by entry to mid-level IT professionals. The purpose of this research was to gain an understanding of contingency planning and to explore what the components of a contingency plan are and how they are used. A contingency plan provides procedures on how to recover IT services in the event of a disruption [6]. In the event of a disruption, we will cover when a contingency plan should be implemented and how to overcome such events.

3.2 Research Approach

Various IT professionals agreed to participate in research efforts and were surveyed to assist in data collection. The surveys that researchers compiled included questions geared toward gaining insight on senior-level IT professional experiences with contingency planning as it relates to creating, implementing, and/or reviewing individual plans. The questions included in the surveys were based on the guidelines and recommendations from the NIST Contingency Planning Guide for Information Technology Systems [7].

3.3 Sampling Method

Each survey was completed by a senior IT professional that was currently or had previously been employed by popular IT companies or entities. The companies or entities where professionals were surveyed included but were not limited to IT professionals employed within government sectors, healthcare, and retail. Research

efforts took place over the course of 2 weeks with a total of 100 professionals surveyed. Within this timeframe, the participants were provided a four-question survey. The questions are as followed:

1. “Do you know what your company’s contingency plan is and what it consists of?”
2. “What is the likelihood that the contingency plan will be used?”
3. “When should a contingency plan be implemented?”
4. “Once a system/service disruption has occurred, what steps are taken to overcome these events?”

3.4 Data Collection Method

Over the course of 2 weeks, the research was conducted in person, online, or by phone surveys. Conducting in-person surveys allowed for interaction and surveying of multiple IT professionals at one location. The online and phone surveys were made available to individuals that were unable to physically be present or those that had other priorities during the research period. The data collected was based on each individual professional’s experience in creating and/or executing contingency plans within their current or previous work environments.

3.5 Data Analysis Method

Data was collected in a qualitative approach. There were no numerical or statistical data to be provided for the research that was administered. The results from each survey were thoroughly examined and grouped together by each company, followed by each individual’s recorded response. Each participant provided written consent to participate in the conducted surveys. The identity of each participant, as well as the identity of each company, was kept confidential and remained protected. All data collected was used solely for the intent of this research.

4 Results

Our investigation of contingency planning led us to a few general discoveries. We found that “most contingency plans are rarely carried out as they are detailed on paper.” [3] This is important because there are many different possible scenarios each with different variables which by default makes every contingency plan a candidate for repeated testing. Most efficient contingency plans test check for vulnerabilities and/or faults in the process as well as any other inefficient or

unnecessary processes. Results reveal that only after thorough testing, contingency plans are needed.

The results have also shown us that while there may be many strategies used to test contingency plans, not all of them count as what are known to be “true tests”. However, they still serve to roughly gauge what policies within the plan need to be updated. Additionally, testing a contingency plan opens the floor for all participating parties to have a discussion surrounding the rehearsal of their part. This then allows for a clearer understanding of roles and responsibilities. The results show that controlled testing is immersive; in addition to this benefit, the testing moderators can actively alter the variables of the scenario which ultimately makes for more efficient testing.

Finally, we discovered that contingency plans had greater success rates when organizations administered multiple “test runs” of each contingency plan component.” [3] If the failure to update these plans was ever to occur, the organization, its information, and various resource changes could decrease the bandwidth to properly handle an incident. This could ultimately result in significant damage to the organization.

5 Discussion

Whenever an organization reviews its strategies, it should adapt over time. Improving plans and rehearsing the revisions are critical to contingency planning. “Each time an incident occurs, the organization should do a detailed review of the lessons learned” [3]. This includes a thorough evaluation of all results.

Additionally, the long-term objective is to implement any discovered changes into an improved set of plans. While doing so, this provides opportunity for constant comparison and evaluation of previous steps from the older plan. In theory, an organization should continue to move forward and improve its contingency plan process so it can strive for an even better outcome.

“Typically, planning for future events is a responsibility reserved for managers in the IT department.” [3] In order for your contingency plan to be considered plausible, it must be supported by the information security department. There are some instances in which organizations have been mandated by law to have contingency plans in place even though it is recommended to not always prepare for every unexpected instance. It is highly recommended that when writing a contingency plan, center it around these four points: business impact analysis, incident response, disaster recovery, and business continuity. A successful contingency planning practice is to have four teams involved with the process. These teams are the: CP, IR, DR, and BC teams.” [3].

Finally, organizations have the choice to either create and/or develop three main planning aspects, or they can choose to individually create three separate elements with intertwining protocols that help with continuity.

References

1. Author Amanda Athuraliya Amanda Athuraliya is the communication specialist/content writer at Creately, Amanda Athuraliya Amanda Athuraliya is the communication specialist/content writer at Creately, A. Athuraliya, Amanda Athuraliya is the communication specialist/content writer at Creately, and View all posts by Amanda Athuraliya →, “What is a Business Contingency Plan: A Step-by-Step Guide,” Creately Blog, 19 Sep 2019. [Online]. Available: <https://creately.com/blog/business/business-contingency-plan-templates/>. Accessed: 30 Apr 2020
2. K. Hughes. How to Make a Contingency Plan, *ProjectManager.com*, 30 Mar 2020. [Online]. Available: <https://www.projectmanager.com/blog/contingency-plan>. Accessed: 30 Apr 2020
3. M.E. Whitman, H.J. Mattford, *Management of Information Security* (Cengage Learning, Boston, 2019)
4. Yin Li, Yiwen Shi and Chen Li, *Constructing IT contingency planning based on RUP model*, 2008 IEEE International Conference on Service Operations and Logistics, and Informatics (Beijing, 2008), pp. 1017–1022
5. J.A. Schafer, *Policing 2020: Exploring the Future of Crime, Communities, and Policing* (U.S. Dept. of Justice, Federal Bureau of Investigation, Washington, D.C., 2007)
6. M.E. Whitman, H.J. Mattford, *Management of Information Security* (Cengage Learning, Boston, 2016). [E-book] Available: Mindtap|Cengage. Accessed 27 Apr 2020
7. M. Swanson, A. Wohl, L. Pope, T. Grance, J. Hash, & R. Thomas. Contingency planning guide for information technology systems recommendations of the national institute of standards and technology, NIST special publication / no. 800, Part 34 (2002) ALL [Online]. Available: <https://norfolkstateu.on.worldcat.org/oclc/109238742>. Accessed 27 Apr 2020

Smart Low-Speed Self-Driving Transportation System



Zhengkong Wang and Bowu Zhang

1 Introduction

The recent advances of self-driving vehicles have drawn great attention from both academia and industry to its applications in various aspects of our lives [1–3]. In this chapter, we are going to introduce a self-driving delivery/monitoring system and investigate market potential of using it in commerce. The proposed self-driving system is composed of low-speed self-driving cars, and a central control unit that can be customized for various purposes. While there exist similar products in the market, for example, Starship Technologies [4], this chapter focuses on the use of such a self-driving system in closed road environment such as school/company campus, office buildings.

Mission We would like to develop an application of self-driving cars to provide automation experience on transportation and patrolling within a closed facility for small organization and end users.

Vision We would like to create valued solutions to ease human labors and improve working efficiency.

The rest of chapter is organized as follows. Section 2 describes the basic idea and the functions of the self-driving system. Section 3 presents two use cases which illustrates more details of system architecture. Section 4 briefly introduces the technologies used in the system, similar products currently in the market, and an estimate of cost and suppliers. In Sect. 5, we analyze the strengths, weaknesses,

Z. Wang · B. Zhang (✉)

School of Computer Science and Mathematics Marist College, Poughkeepsie, NY, USA
e-mail: zhengkong.wang1@marist.edu; bowu.zhang@marist.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence. https://doi.org/10.1007/978-3-030-70873-3_68

943

opportunities, and threats (SWOT) of the proposed system. We also discuss the target customers and marketing strategies in Sect. 6. The chapter is concluded in Sect. 7.

2 System Basics

The system consists of self-driving cars and a control center which can be customized for various applications. Each car will connect to the control center through one or more types of mobile networks, such as 4G, future 5G, or Wi-Fi network. They will drive themselves in a closed road environment such as a campus or a factory and through buildings like an office building in a low-speed that is a little bit faster than walking speed. Vibration proof suspension keeps it having a smooth ride, while many security features make it safe and strong.

3 Use Cases

3.1 *Delivery System*

The proposed system can be easily adapted for a smart delivery system. For the delivery purpose, there should be multiple cuboid-shaped cabins stacked on top of the car. They could be assembled in various combinations with modules in multiple sizes locked with an e-lock. The lock will make sure no one else could open it until it reached the destination. If we take the example of deploying the system on a campus, we can visualize a very popular use. User around the campus could send a request with a smart phone app, website, or call the control center directly. Control center will schedule the trip and one of the cars will come to your location and pick up the thing whether if it is some big boxes of documents, some small mails or even a meal box. The car will take it to the destination you ordered. Sender will identify them self with code or ID card. Receiver will input a code sent by the system or swipe ID to confirm identity as well. After completion, you will be charged for this trip. If the car is installed with human seat module, it could send people with disabilities to their destination as well (Fig. 1).

3.2 *Patrol System*

Another way to use the proposed system is patrol. Replacing human security guards driving SUVs with more of these electricity powered cars will dramatically decrease cost and improve environment. Each of the cars will equip basic night-



Fig. 1 Food delivery system

vision cameras and microphones. Basically, it will be similar to securities watching live video stream in the control center. It could also install software and hardware upgrades in the future for automatic detection for someone yelling and calling for help. This function will be performed at the same time of delivery. Otherwise, when there is no delivery mission, they will run as full patrol mode moving around the location.

In addition, there exist a significant number applications we can use the proposed system: helping hand when hosting events, maintenance tool storage, medical equipment transport, sticking advertisement on car body, add cleaning module to swipe the ground, etc. Sky is the limit.

4 Technologies and Cost

4.1 Key Technology Involved

The project has several key components: car design and production, car control software development, control center software development, and user training. The most difficult part will be the control software of the car. It will include some degree of environment recognition and interaction. In order to be able to not relying on stable network connection, the car will have to make simple information collection and decision by itself, performing as a self-driving car. It will remember the map of the facility and locate itself with GPS and assistance signs like QR code stick to poles by side. When wanted, a remote-controlled mode commanded from the control center is also available. All these functions will guarantee a safe operation against unexpected situations and accidents. Self-driving and low-speed will prevent

it from running into person and properties. Manual monitoring will prevent it from active abuse of the car and the system. Actual persons are the backup of the automated system.

While there exist similar products in the market [4], for example, community delivery using drones, this chapter focuses on the use of such a self-driving system in closed road environment such as school/company campus, office buildings.

4.2 Cost, Budget, and Suppliers

The cost of starting this project will cover the prototyping, customer beta testing and mass production of cars when market is ready. It will include software development, management and advertising as well. Since this project requires physical product, the budget will be short before we could actually deploy the system to customers. Thanks fully the system is achievable using unpatented public technologies from GitHub and open-source software. The car is easy as well with 4 small motored wheels, which would be similar to a robot vacuum. Hardware side, we can make prototype on site and collaborate with factories in China to reduce cost and increase quality and manufacture efficiency. Software side, we could build our own team on site, if failed to form a good team, we can also outsource this part to other teams or companies. Management and marketing team will monitor and arrange all resources required.

5 SWOT Analysis

5.1 Internal Strengths

The car used in the proposed system is safe and simple to design, it is basically just a big robot vacuum cleaner plus some cabinet on top. The control system does not need to be perfect at the beginning since human is still able to monitor and control it. It has wide variety of use cases. It looks cool and relatively cheap and convenient that makes people want to try it. Upgradeability and creativity, customers can install anything to our product they want. It is customizable to fit the need of every customer in any use cases.

5.2 Internal Weaknesses

The self-driving car often comes with long time R&D process that relies on outside investments and crowdfunding. Mass production of a robot this size may require another big amount of investment for more efficient production line to lower the price.

5.3 Outside Opportunities

The market on self-driving vehicles is fairly new and there are opportunities for early birds. We are targeting businesses, and that will make our revenue stable base on the B2B model, end customers could spread and advertise this futuristic product and reduce advertising costs.

5.4 Outside Threats

A potential problem may come from traffic regulations though our system is designed for closed environment. People walking around may worry about safety of this product. Then, big companies like Google, Amazon, Alibaba, DJI, and Tesla have already realized the market in this area, they will develop their products faster with more budget in the near future, we have to plan on how to get over that situation with end customers and contracts with other companies. Social counterviews could also be a problem, like data privacy and cyber security concerns. Also, it is taking the job of lots of paid persons who may go against the idea of introducing automation. Some worker may even damage the system secretly to go against these robot cars. On the other hand, humans are already doing the same job, it is challenging to make our system appeals to be cheaper, more convenient, and more fun to deploy comparing to keeping old jobs and systems.

6 Market Analysis

6.1 Target Customers

Our product targets majorly on businesses, like Business facility owner, organization owner, schools, colleges, office mansion, factories, etc. This will be our main revenue. Our robots could also be purchased by individual end users, in a smaller scale. They will help as a part of advertising. Since this is a front edge futuristic product, many people would interest in it and have a look or try. We have included examples of target customers in real world:

- Post officers on campus: Usually, mails and packages were picked up by individuals, but when there are items sent to specific department or buildings, a mail man needs to deliver mails in person to each office building. Using our product, the post office can use self-driving vehicles to deliver mails and packages, which saves time and human labor.
- School Cafe and Dining Service: Employees and students can place order online and the food will be delivered by our system.

- Security Office: Instead of walking through the whole campus or entire building, security officers can remote control self-driving vehicles, particularly, to see the blind spots which cannot be seen by surveillance camera.

6.2 Marketing Assets

We will advertise to schools and companies who would like to use our product for a trial. This is also part of product improvement and bug-fix period, as well as a way to show case the use of our product. We would also reach for crowdfunding which will not only get us financial support, also a way to improve end user popularity and engagement. Online influencer's voice as well as TV media programs are additional ways to advertise. We can also send beta versions to content creators for a try. In addition, we can participate tech exhibition to enlarge our business impact to different groups of participants, even giant companies.

7 Conclusion

In this chapter, we discuss a commercial application of self-driving vehicles. We investigate use cases, explore cutting-edge technologies, highlight prototyping activity, and identify risks and needs for risk management. In addition, we also discuss target customers and marketing strategies for the proposed application.

References

1. W. Wang, D. Zhao, Extracting traffic primitives directly from naturalistically logged data for self-driving applications. *IEEE Rob. Autom. Lett.* **3**(2), 1223–1229 (2018)
2. L. Liu, H. Li, J. Liu, C. Karatas, Y. Wang, M. Gruteser, Y. Chen, R.P. Martin, BigRoad: Scaling road data acquisition for dependable self-driving, in *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services* (2017)
3. E. Ohn-Bar, M.M. Trivedi, Looking at humans in the age of self-driving and highly automated vehicles. *IEEE Trans. Intell. Vehic.* **1**, 90–104 (2016)
4. Starship Technologies, Starship Campus Delivery Service With Robots. YouTube (2020). <https://www.starship.xyz>

Are Collaboration Tools Safe? An Assessment of Their Use and Risks



Cquoya Haughton, Maria Isabel Herrmann, Tammi Summers, Sonya Worrell, Samuel Olatunbosun, and Taiwo Ajani

1 Introduction

In the latter part of 2019 a pandemic, called COVID-19, dramatically changed the social and professional and, perhaps, the political landscape. In the United States, many states and organizations took action to slow the spread of the deadly virus by instituting stay-at-home orders and mandatory telework. Therefore, as more people began to work, study, and isolate themselves at home, the demand for virtual telecommunications increased.

One of the primary virtual demands was the need for collaboration tools to provide business continuity for organizations that do not commonly practice virtual operations. A collaboration tool is an application software that facilitates information sharing and communication among people in distributed geographical locations.

Zoom, Skype, GoToMeeting, and Google Hangouts were some of the more popular solutions. Each tool with its unique security features, communication capabilities, and functions requires a deeper look to inform consumers on the associated risks and vulnerabilities that present opportunities for threat actors. The four tools discussed in this paper are:

C. Haughton · M. I. Herrmann · T. Summers · S. Worrell · S. Olatunbosun (✉)
Department of Computer Science, Norfolk State University, Norfolk, VA, USA
e-mail: c.haughton@spartans.nsu.edu; mherrmann@nsu.edu; t.summers@spartans.nsu.edu;
s.m.worrell103094@spartans.nsu.edu; sbolatunbosun@nsu.edu

T. Ajani
Department of Computer Science and Information Systems, Ferrum College, Virginia, VA, USA
e-mail: tajani@ferrum.edu

1. Google Hangouts.

- (a) Google Hangouts allows you to exchange text messages, pictures, and videos with anyone you know who has a Google account. The application works with your Google account, allowing you to communicate with your Google contacts through text messages, video chats, and voice calls.
- (b) In 2013, Google added the ability for the Hangouts app to also serve as your default text messaging (SMS) app, replacing the messaging app that also comes preloaded on your phone. Now all your messages – Hangouts and SMS text messages – are in one place, and no other messaging service such as Facebook Messenger or WhatsApp can offer the same.
- (c) If you are on the go, you can start a Google Hangout session on your computer and seamlessly continue it on your mobile device.

2. GoToMeeting.

- (a) GoToMeeting is a web-hosted service that provides features such as online meeting, webinars, desktop sharing, and video conferencing to enable users to meet with other users and clients in real time via the Internet [1].
- (b) GoToMeeting is compatible with iOS, Windows, and Android computing devices.

3. Skype.

- (a) Skype is one of the first online communication collaborative tools.
- (b) Features includes audio, HD video calls, chat feature, and screen sharing.
- (c) Skype offers worldwide calling and video calls with multiple users. The intended audience are those that want to connect with those that they do life with and work with.
- (d) The app is built for both one-on-one and group conversations and is available on mobile, personal computers, Xbox, and Alexa.

4. Zoom.

- (a) Zoom aims to help businesses and organizations bring teams together in a frictionless environment.
- (b) It is a cloud-based video and audio telecommunications platform and a state-of-the-art software-based conference room solution [2].

The preceding sections identify these four prominent collaboration tools, reflect on expert opinions on the tools' security posture, and lastly, offer recommendations on how consumers and organizations can improve security while using the tool.

2 Literature Review

Moszkowicz, Duboc, Dubertret, Roux, and Bretagnol wrote about medical students' experiences in surgical departments regarding adjustments as a direct result of

COVID-19 and the cancellation of nonurgent surgical activities. In conjunction with those cancellations, universities and businesses closed their doors and asked nonessential personnel to stay home. Moszkowicz et al. detailed efforts to continue daily medical education for surgical students confined to their homes [3]. According to the authors, Google Hangouts was implemented to continue learning and adhere to safety guidelines. The purpose of this method was to compensate for the withdrawal of clinical lessons performed daily on the board in the surgery department [3].

Many universities and workplaces use a form of Google's email services and having a Gmail account is an easy way to access Google Hangout. Moszkowicz et al. found that the Google Hangouts app was available to all at no cost [3]. The only additional purchase necessary would be a webcam and a microphone, but most laptops, cell phones, and tablets have those items built in. They found that Google Hangouts was very similar to regular lessons because it allows the administrator to set up appointments and send out an e-mail with all the information [3]. Up to 10 students can access the live lesson at the same time, see and hear the administrator, and ask questions that are audible to the entire group, and there is no time limitation on the live stream [3].

Anatomical and medical education traditionally employs schooling methods such as physical lecturing and blackboard chalk drawings, inside a classroom for an extended period of time [3]. With the development of the COVID-19 pandemic, it is necessary to rethink the classroom and, more specifically, education. Google Hangouts has provided a way to implement blended learning. Blended learning is defined as the combination of conventional face-to-face learning and asynchronous or synchronous e-learning [3]. This avenue of blended learning reserves face-to-face class time to student-centered activities that enhance active learning and the online approach for introducing the topic [3]. The onset of the pandemic has changed our workspace and classrooms, and Google Hangout provides a way to stay connected and continue education.

3 Is Google Hangouts Safe, Secure, and Private? | Tech Boomers

Google Hangouts encrypts your information and conversations to protect your safety and privacy [4]. As long as it is utilized to communicate with familiar and trusted people, it can be relatively safe using all of the communication options on Google Hangouts. Google Hangouts also encrypts conversations to keep them private. Text chat messages on Google Hangouts are also archived, much the same way that messages on various social media platforms are, so while your messages are private, Google still has a record of them [4]. The main thing to check for when choosing the right application to use is the encryption setting. End-to-end encryption means your private chat messages are scrambles, and only the sender and the receiver of

the messages have the “keys” to read them. The app security feature ensures that no one besides you and the recipient can decipher the messages [5].

On the flip side, Google does not use end-to-end encryption; therefore, if the government requested access to your Google Hangouts conversations, Google could allow them access [4]. Google Hangout utilizes “in transit” encryptions, and this means that they are only encrypted between your device and Google’s servers, and once on the server, Google has complete access to them [5]. Google Hangouts is private, and when you send messages or have a conversation with someone on Google Hangouts, it will only be visible to you and the other person/people in the conversation [4]. Other users will not be able to see your conversation unless you add them as a contact and invite them to join the chat.

There are four tips to staying safe using Google Hangouts. The first is not to communicate with anyone you do not know [4]. Add or accept requests from people you know and trust such as co-workers, family, and friends. The second tip is to prevent anyone else from accidentally accessing your account by signing out when you are done using Google Hangouts [4]. When using a shared/public computer, make sure to sign out of all accounts and delete the history on that computer, which minimizes the risk of someone else using your account without your permission. The third tip is to secure your account with a good password [4]. Having a strong password helps to protect your account in case someone tries to hack into it. It is also helpful to change your password every 90 days. The final tip to help you stay safe is if anyone happens to be bothering you on Google Hangouts, block them [4]. Block anyone who bothers you or someone you may have added by mistake so that they no longer pose a threat to you or your communication platforms.

4 Google Cloud in the Era of the Pandemic

Google Cloud CEO Thomas Kurian illustrates how the way we work, communicate, and learn have changed with the onset of the pandemic and how Google is smoothing the transition to no-contact environments. All over the world, businesses and users depend on Google Cloud to help them stay connected and get work done. More businesses rely on connecting an at-home workforce to maintain productivity. As a result, there has been an unprecedented surge in the use of Google Meet.

The advanced features in Google Meet (GM) are now free to all G Suite and G Suite for education customers globally. Over the last few weeks, GM day-over-day growth surpassed 60%, and as a result, its daily usage is more than 25 times what it was in January [5]. The MACIF Group, a leading French mutual insurance provider, was able to ensure business continuity and maintain the link between its employees with G Suite, already deployed to more than 8000 employees [5]. MACIF staff shifted from in-person meetings to more than 1300 Google Meet video meetings daily, and the use of collaborative virtual rooms facilitated important human contact and responsiveness in an unexpected period of remote work [5].

During this transition to remote work and learning in response to the pandemic, many are looking to build their skills and increase their knowledge while at home and to help Google by offering their portfolio of Google Cloud Learning resources at no cost until April 30 [5].

According to Kurian, educational institutions have been particularly impacted by the coronavirus, and initiatives are being undertaken to support them, ranging from providing free content and educational tools to supporting distance-learned initiatives that help educators to continue teaching students at home. For example, Google Classroom was available to 1.3 million students in New York City in order for them to continue their school year virtually at home [5]. Khan Academy supports 18 million learners per month before the crisis and since the school has closed it has seen record growth across all metrics, 20 times the normal [5].

Several companies have given their testimonial regarding successes with Google applications including a Korean gaming company NetMarble that discussed how Google Hangouts helped them make the company-wide smooth transition to working from home. The company stated that “with video conferencing through Google Meet, collaboration via Google Hangouts, and all data accessible on Google Drive, there’s really no difference when working from the home or the office” [5]. Google is working with state agencies like the Oklahoma State Department of Health on solutions for medical staff to engage remotely with at-risk people who may have been exposed to the coronavirus [5]. The department deployed an app that allowed medical staff to follow up directly with people who reported symptoms and directly affected citizens to testing sites [5]. Google Hangouts has helped organizations continue their daily operations and allowed schools to continue providing education during this pandemic, which allows our society and economy to move in a slow but steady progression.

5 Cyber Threats Related to the Coronavirus and Security Management

Rennie and Mathieu discussed the various cyber threats and how the security team addresses the pandemic-related cyber threats. Through a partnership with external intelligence communities, personal and professional groups (like IT-ISAC), and their own internal research group, indicators of compromise (IoC) – IP addresses, domains, hashes – were gathered for further investigation and analysis. Threats like phishing, denial-of-service (DoS) attacks, ransomware, SMS phishing, and other malwares were found. Malwares such as Maze and NetWalker that have targeted vaccine testing facilities and hospitals were also observed [6].

According to the authors, deploying an incident response plan during this time of crisis, when remote communication and collaboration have increased tremendously, is the key to mitigating these cyber threats. The team’s global Computer Security Incident Response Team (CSIRT) works 24/7 together with their Threat Intelligence

and Vulnerability Management Team who can assemble as one unit at lightning speeds when an incidence requires immediate investigation [6].

Other companies collaborate with video conferencing providers such as GoToMeeting to address software vulnerabilities and resolve the potential issues that could have been exploited by criminal hackers [7]. For instance, Swascan, a cybersecurity outfit, was able to identify some vulnerabilities that were shared with the video conferencing tool software PSIRT through the responsible vulnerability disclosure, thus forcing GoToMeeting to decommission the server that could have caused potential security breaches [7].

The two companies, GoToMeeting and Swascan, merged resources and expertise to address the potential doorways to hackers [1]. Swascan also credited the success of this endeavor with the succeeding security-enhancing partnership with GoToMeeting and various players. According to the cybersecurity company, the key to any responsible vulnerability disclosure activity is the collaborative efforts between cybersecurity providers and the service providers.

6 Skype and Microsoft

The key features of SKYPE are voice and video calling. Users can call landlines and mobiles from anywhere in the world via mobile, PC, Xbox, and Alexa. SKYPE offers messaging and two usage plans. The service is free for those who send messages and those that have audio and video calls with groups of up to 50 people (Skype to Skype Calls). The only downside is that no emergency calls can be made with Skype. Affordability is another feature that's attractive because Skype allows users to pay as you go [8]. Skype to Skype has the built-in feature to record Skype-to-Skype calls and store these files on the Microsoft servers for up to 30 days. Users are alerted when calls and screen sharing occur as both are recorded via host request.

Microsoft acquired Skype in 2011 for \$8.5 Billion [9] and adheres to Microsoft's privacy statement. Although Skype is not specifically mentioned in Microsoft's privacy statement, however, it can be assumed that the platforms essentially share the same Microsoft's privacy policies. Each profile consists of a profile picture, username, password, email address, location, birth date, and contact list. Microsoft records who you call, time and duration of calls, chat history, sent and received files, phone numbers called, and activity status [9]. Microsoft uses this information for targeted advertising, personalization, research, and development. This data is also shared with Microsoft affiliates, subsidiaries, and vendors. In regard to the security measures of Skype, Microsoft allows the user to manage their online privacy settings for some, but not all of the data is private. Skype does allow the user to be discovered or not through its "discoverability" feature. There are security measures that Skype offers for user logins. One would be the two-factor authentication, which is a great disincentive for potential hackers. This process requires the user to enter a one-time code sent via email, text, or app in addition to the password. The other security log-in option is usage of a security key which is utilized via a USB device and

must be plugged in when logging into one's account. Skype does not use end-to-end encryption. This means Microsoft can view every message, file and call. Microsoft has Skype set up for encryption, however only between the user's device and Microsoft servers. Data is only encrypted once it reaches the Microsoft server and leaves the door open to the data prior to reaching the server.

7 Security

Despite this, security risks still exist. According to Cimpanu [10], it is assumed that a lot of abandoned Skype accounts are still out there. This leads to hackers being able to use these accounts to spread malware, similar to the 2019 occurrence when the Rietspoof malware was spread primarily through Skype spam. This malware was a Trojan designed to infect systems so it could download more intrusive and potent malware. Skype also exposes IP addresses, which is done through a single command in the Windows command prompt listing all of the TCP connections to your device.

With regard to the usage of Skype in the healthcare industry, there are several red flags [11]:

- Skype has the rights to data that is transmitted and can review the data at its own discretion,
- Data can be monitored via digital wiretapping, if required by government organizations,
- Skype does not provide audit trails or notifications in case of a breach, which can therefore go completely undetected,
- There is no specific service level availability for Skype, so quality cannot be guaranteed.

8 Settings for Securing Zoom

A UC Berkeley article states that UC Berkeley's Zoom platform may only be used for certain classifications and that it may not be used for anything that includes social security numbers, financial account information, and controlled data. UC Berkeley also acknowledged that they use Zoom HIPAA, which transfers HIPAA data [12]. This is an interesting point to consider that UC Berkeley trusts its Zoom platform to transmit data such as IT resources, IT security information, staff and academic personnel records (including employee ID), and medical devices supporting diagnostics [13]. This establishes that Zoom is incorporated into reputable organizations and IT infrastructure and is trusted to exchange sensitive information on the specialized HIPAA Zoom.

UC Berkley went on to share best practices to maintain secure and safe meetings. Some of these measures include (1) keeping the platform up to date with the latest versions, (2) not sharing meetings publicly but manage distribution and access through Zoom's setting features, passwords, and avoid "Join before host," (3) using a waiting room which allows the host to admit users individually and once the host is ready, and (4) removing unwanted or disruptive participants and placing participants on temporary audio and video holds.

9 Zooming to Conclusion Cybersecurity

The United States changed in the face of social distancing measures. As a result of the disruption, the citizenry has had to adjust in several ways from personal habits to professional routines and board room meetings. Trollope in an article titled, "Beyond the noise – 7 reasons – it's safe to run Zoom," acknowledged that his company uses the Zoom platform and had significant investments in the company. The story has since been deleted [14]. According to Crisler, in spite of its software vulnerabilities, weaknesses in its encryption technologies, and the lack of oversight over its security controls [15], Zoom is still worth it. Crisler indicated that Zoom uses a nine-digit conference room number. The key here is that the ID only uses numbers. With this format, there are a total of 1,111,111,110 potential combinations that can be randomly guessed by a smart computer within 2 weeks or in under 2 seconds by a special password cracking tool [15]. The fix to this is the host utilizing a unique password. This lowers the chance of someone hijacking the conference. Crisler [15] shares similar security features previously stated by UC Berkeley like the use of the waiting room.

Zoom acted very quickly to assist educational institutions amidst the crisis but found itself under public scrutiny for perceived security flaws. In February of 2020, Zoom offered its collaboration tool to schools for free so that they could continue teaching. Over 90,000 schools across 20 countries took the company up on their offer along with millions of others who were looking for ways to connect through social distancing rules [16]. The article further explains the actions taken by Zoom to address their security issues which included the following: (1) The Zoom platform was primarily created for enterprise-level organizations like financial services, government agencies, universities, healthcare organizations, and telemedicine practices that have robust IT support. (2) Zoom removed the Facebook SDK in its iOS client and reconfigured it so that it doesn't collect unnecessary device information. (3) The update of Zoom's privacy policy to make it more transparent to users clarifies that Zoom doesn't offer end-to-end encryption of its meetings [16].

9.1 Research Findings

Of the four tools, GoToMeeting is the only tool that confirms end-to-end Secure Sockets Layer (SSL) and 128-bit Advanced Encryption Standard (AES), and no unencrypted information was ever stored in the system. Many of the tools studied, outside of Skype, were not created to support sensitive information transmissions; they were created to support private communications. However, this does not absolve the companies from having high security standards to protect the privacy and integrity of all communications transmitted on their platforms.

Tools like Skype, Google Hangout, and Zoom were created to cater more toward social interactions and did not receive the type of scrutiny normally rendered through acceptance testing, which most organizations require to prove strong security standards. Additionally, the increase in uneducated consumers on these collaboration platforms magnified vulnerabilities and gave opportunities for hackers to take advantage of weakly secured communications. Many of these tools, when used for their intended purpose, are safe, but when used for business or other more sensitive communications, extra attention needs to be paid to utilize available security features in the tool.

9.2 Recommendations

In light of the essential need for collaboration tools during the pandemic, these companies have an obligation to publicize best security practices of their tools through a wide range of correspondence, including e-mails and the prominent display of best practices on the site's main page. There is also an opportunity for tools to reconfigure the "introduction" upon entering the tool which would offer a "mandatory" tutorial for the consumer to walk through the security features. While users may skip through these tutorials, it is another way to have users acknowledge the features exist. Educational institutions should also require online training sessions for teachers to familiarize themselves with how to secure their virtual meetings. In the case of zoom, they would learn how to remove participants, mute participants, and control their meetings and security settings.

Consumers should also research and review the purposes, security features, and access controls before implementing such in their organizations. For meetings that discuss sensitive topics, the most secure platform out of the four presented here is GoToMeeting due to its end-to-end encryption. Churches and schools who may need a more economical platform and not concerned so much with encryption but with privacy should look at the ease of use and the security features offered. They should also become familiar with the security features within the tools. Features such as password protection, waiting rooms, and room access are all features that improve the integrity of the meeting and the privacy of the call.

10 Conclusion

Collaborative tools, once considered a tool of the future, became tools of the present during the pandemic. These tools allow society to continue forward progress in learning, innovating, and streamlining processes. Collaboration tools help maintain communication between people worldwide and promote productivity.

The direct results of the pandemic response were the limitations on the number of people that can be in a space at a given time and also the minimum distance between each individual in that space. The limitations set forth by the government may be considered a significant impediment to business continuity, and as a result, many businesses, organizations, and schools closed their doors. The availability of collaboration tools such as Google Hangouts, Zoom, GoToMeeting, and Skype bridged the gap virtually, making geography less an obstacle.

Google Hangouts, Zoom, GoToMeeting, and Skype offer online options to stay connected for work or personal use. Although these tools were present before the pandemic, they had fewer active accounts. Each tool saw an exponential increase in users during the pandemic, which led to many users not knowing how to protect themselves within the tool. This left their accounts and online sessions vulnerable to malicious activity.

Anything done online comes with risk and vulnerabilities, but it is up to users to mitigate and minimize as many risks as possible by following security guidelines.

Each collaborative tool is taking a proactive approach to encourage their users to follow information security guidelines while using their tool in order to keep their sessions protected and safe.

Not only does the use of collaboration tools have its benefits, but also it introduces new and unintended consequences such as novel security threats to which organizations must respond. Each organization, school, and business' needs are different; therefore, the best collaboration tool for your organization will reveal itself after researching and evaluating the various platform's functions, capabilities, and security infrastructure.

References

1. R. Chiesa, P. Lezzi. GoToMeeting and Swascan collaborate to proactively address software vulnerability. Swascan, 9 Nov 2019. [Online] Available: <https://www.swascan.com/gotomeeting-and-swascan/>. Accessed: 24 Apr 2020
2. Zoom. About Zoom, Zoom (2020). [Online]. Available: <https://zoom.us/about>. Accessed 24 April 2020
3. D. Moszkowicz, H. Duboc, C. Dubertret, D. Roux, F. Bretagnol. Daily medical education for confined students during COVID-19 pandemic: A simple videoconference solution, Wiley Online Library, 22 Apr 2020. [Online]. Available: <https://onlinelibrary.wiley.com/doi/full/10.1002/ca.23601>. Accessed: 23 Apr 2020
4. Is Google Hangouts Safe, Secure and Private? 4 Safety Tips. TechBoomers.com, 25 Apr 2017. Online. Available: <https://techboomers.com/t/is-google-hangouts-safe-secure-private>. Accessed: 20 Apr 2020

5. T. Kurian. How Google Cloud is helping during Coronavirus | Google Cloud Blog. *Google*, 31-Mar-2020. [Online]. Available: <https://cloud.google.com/blog/topics/inside-google-cloud/how-google-cloud-is-helping-during-covid-19>. Accessed: 19 Apr 2020
6. P. Heltzel. COVID-19 and tech: New collaboration tools mean new security risks, InsiderPRO, 19 March 2020. [Online] Available: <https://www.idginsiderpro.com/article/3532506/covid-19-and-tech-new-collaboration-tools-mean-new-security-risks.html>. Accessed: 13 Apr 2020
7. Swascan Team. Swascan collaborates with video conferencing provider to proactively address software vulnerability. CISION PR Newswire, 11 Nov 2019. [Online] Available: <https://www.prnewswire.com/news-releases/swascan-collaborates-with-video-conferencing-provider-to-proactively-address-software-vulnerability-300954882.html>. Accessed 25 Apr 2020
8. Skype. About Skype. Skype, 2020 [online]. Available: <https://www.skype.com/en/about/> Accessed 24 Apr 2020
9. D. Gilbert. Is Skype Safe and Secure? What are the Alternatives? <https://www.comparitech.com/blog/information-security/is-skype-safe-and-secure-what-are-the-alternatives/>. [Online] April 1, 2020. Accessed 24 Apr 2020
10. C. Cimpanu. Rietspoof malware spreads via Skype spam. <https://www.zdnet.com/article/rietspoof-malware-spreads-via-skype-spam/> [Online] February 19, 2019. Zero Day. Accessed 23 Apr 2020
11. T. Warren. Microsoft's Skype struggles have created a Zoom moment. <https://www.theverge.com/2020/3/31/21200844/microsoft-skype-zoom-houseparty-coronavirus-pandemic-usage-growth-competition>. The Verge. [Online] March 31, 2020. Accessed 24 Apr 2020
12. UC Berkeley Information Security Office. Security for Securing Zoom (2020). [Online]. Available: <https://security.berkeley.edu/resources/cybersecurity-and-covid-19/settings-securing-zoom>. Accessed 25 Apr 2020
13. UC Berkeley Information Security Office. Information Security Office Data Classification Standard (2020). [Online]. Available: <https://security.berkeley.edu/data-classification-standard#plmoderate>. Accessed 25 Apr 2020
14. R. Trollope. Medium, Beyond the noise- 7 reasons it's safe to run Zoom (2020). [Online]. Available: <https://medium.com/@rowantrollope/beyond-the-noise-7-reasons-its-safe-to-run-zoom-e366721b5a8b>. Accessed 13 Apr 2020
15. V. Crisler. Medium, Zooming to Conclusions. 5 April 2020. [Online]. Available: https://medium.com/@vince_17729/zooming-to-conclusions-20560d9f40b9. Accessed 13 Apr 2020
16. J. Bowles. Diginomica, It's time to stop bashing Zoom (2020). [Online]. Available: <https://diginomica.com/its-time-stop-bashing-zoom>. Accessed 13 Apr 2020

Tourism Service Auction Market for Saudi Arabia



Saad Nasser Almutwa and Sungchul Hong

1 Introduction

1.1 *Tourism in Saudi Arabia*

Saudi Arabia is a large country in terms of geographical area and contains a vast number of tourist places and attractions; there are five places registered so far in UNESCO as historical and tourist destinations [1]. According to the Saudi Vision 2030, the Saudi government will work to establish heritage tourism places and take care of them through the goals stipulated in the vision; additionally, this project aims to double the number of heritage sites registered by UNESCO [2]. Moreover, the kingdom has established commissions to support tourism. For example, the royal commission for AIUla will be responsible for delivering infrastructure in AIUla and completing responsible tourism development projects to make AIUla a world-class tourist destination [3].

Previously, travel to Saudi Arabia for the purpose of tourism was difficult due to visa restrictions. Currently, Saudi Arabia is opening its national tourism industry to tourists who wish to come to Saudi Arabia. For current and future tourist destinations in Saudi Arabia, such as AIUla and Neum (respectively), this movement is in line with the Kingdom's Vision 2030 and aims to both attract tourists and simplify the process of obtaining a tourist visa. Given this wide-open opportunity for tourism in Saudi Arabia, it is clear that there is a strong case for the development and introduction of an electronic market for tourism in Saudi Arabia. The utilization of an electronic market would improve the buying and selling process between tourists

S. N. Almutwa (✉) · S. Hong

Computer and Information Sciences Department Towson University, Towson, MD, USA
e-mail: salmut4@students.towson.edu; shong@towson.edu

© Springer Nature Switzerland AG 2021

H. R. Arabnia et al. (eds.), *Advances in Software Engineering, Education, and e-Learning*, Transactions on Computational Science and Computational Intelligence, https://doi.org/10.1007/978-3-030-70873-3_70

961

and the agents who act as providers of services. This electronic service market will improve communication between the tourist and the service provider (agent) while increasing overall efficiency. Both sides would be able to trade services by using trading software and communication networks.

An electronic market for the tourism services between sellers (service providers) and buyers (tourists) is proposed in this paper. For instance, a tourist often has a list of desired activities for their trip. Each agency has its own list of services they provide; a single agency might not cover all the activities that tourists desire, so a consortium of agencies may be more able to fulfill the entire list of activities desired by the tourist.

2 Literature Review

2.1 Background of E-marketplaces

In recent years, as Internet technology has grown, traders have begun utilizing the Internet as a means of selling goods and services [4, 5]. This electronic form of trading is known as the electronic market, “online trading,” “e-marketplaces,” or “electronic commerce” [6]. As e-marketplaces expand, many benefits of utilizing the Internet in trade have been noted [7, 8]. These benefits include Internet assistance for organizations and delivery services [8]. Additionally, the Internet aids in the exchange of management information and provides support for strategy implementation. The structure of the e-marketplace varies based on business models and the industry of focus [9].

2.2 Consortium

A consortium refers to two or more members collaborating together, as a buyer or a seller, to achieve a common objective within the marketplace [10, 11]. Moreover, Ivanovic et al. [12] presented the explanation of a consortium as a contractual relationship meant to benefit all participants within an investment project. The definition of a consortium within a local government is the arranged compilation of the needs of multiple local government organizations to optimize the purchasing power and available resources of the government [13]. Documentation and management are crucial processes necessary for the development of a consortium [12]; additionally, membership fees, consulting fees, and transaction fees make up the bulk of consortium income in an e-marketplace [14]. On the other hand, consortiums in e-marketplace serve different functions for both buyers and sellers. Consortiums benefit buyers by reducing the cost of products when compared with large sellers online [14]. For example, an agent does not need to provide a lot of services;

whenever an agent needs a service that they do not provide, they may outsource this service via a consortium. In reference to sellers, consortiums increase the sales of their products [14].

2.3 *Clearing House*

The establishment of rules and regulations, as well as the enforcement of contracts, is the responsibility of the clearinghouse [15]. In the market, buyers and sellers must have accounts with the clearinghouse before accessing the services provided [16]. The clearinghouse serves as a substitute for direct counterparty relationships between buyers and sellers by acting as an intermediary [15, 16]. Additionally, a clearinghouse takes on the role of a counterparty to both sides within the market [15].

As mentioned above, it ensures that all parties in the marketplace perform their duties equitably as directed by their contracts [16]; however, the clearinghouse can only provide these services to those who already have an account with the clearinghouse [15]. An agent may encounter many potential problems, such as an out-of-order bus or bad weather; despite these potential complications, agents must provide their service to their best efforts.

However, in this case, the clearinghouse function is different from the traditional clearinghouse function. The proposed clearinghouse will deal with intangible services, not products, and work to guarantee the services are provided. Additionally, the clearinghouse will work to enforce rules. In this model, these potential problems are excluded due to the model complexity.

2.4 *Set Cover Problem*

A set cover problem is defined as, “a set of sets whose union has all members of the union of all sets. The set cover problem is to find a minimum size set” [17]. A set cover problem is classified as one of Karp’s 21 NP-complete problems and is recognized as a typical problem in computer science [18]. Therefore, a heuristic algorithm is utilized in this proposal. The definition of set cover problems can best be explained using a set of elements. For example, the universe U is given n elements and a collection of subsets of U , $S_1 \dots S_k$, with positive costs specified, the minimum set cover problem is to find the minimum cost of elements of the sets whose union is U [19].

Solving set cover problems presents several possible complications, such as set redundancy and infeasibility of solutions [20, 21]; for these reasons, set cover problems are heavily considered to be NP-Hard. To solve set cover problems, a heuristic algorithm is utilized to produce possible solutions that would generate favorable outcomes in practice. Set cover problems have been applied for various purposes,

such as crew scheduling on railways and extended transit issues with transportation companies [22]. Despite the possible difficulties of set cover problems, there are countless instances in which the problem-solving process has yielded practical and effective solutions. In this paper, there is a set of trips and the consideration of the lowest costs during pairing; the set of services provided by the agents that includes all requested services from the activity list of the tourist is used as set cover.

3 The Proposed Tourism Service Auction Market

The proposed electronic market for tourism works as follows: a tourist submits his or her wanted activity list to the electronic market; then, the market system tries to match this list with the collection of lists of service providers. If a service provider cannot provide the requested service, a consortium of services from many different service providers will be formed with the lowest service price. The service fee transaction will be guaranteed by a clearinghouse. The tourist should have an account with an adequate amount of money, and tourist agents must make sure the promised services will be provided. This electronic market for tourism is depicted in Fig. 1.

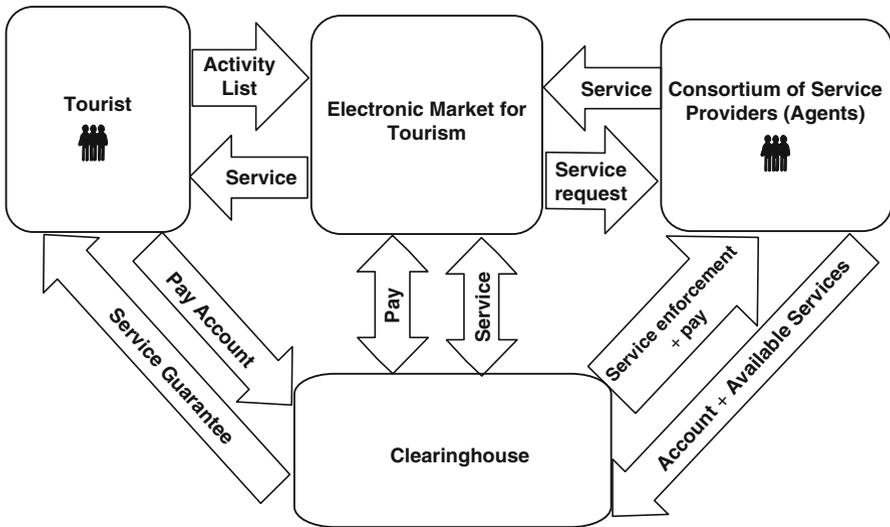


Fig. 1 The proposed electronic market for tourism in Saudi Arabia

3.1 Market Model (Math Model)

An automated, electronic, agent-based auction is applied to a tourist activity service market. A tourist has a wanted list of activities. There are many tourists $\{t_1, \dots, t_m\}$ with tourist activity lists (TA). The tourist j will have $TA_j = \{a_1, \dots, a_n\}$. There are many tourist agencies $\{a_1, \dots, a_q\}$ with service activities (SA). The agent i will have service provider activity list $SA_i = \{a_n, \dots, a_m\}$. This market first aims to match the tourist's activity list with the agents' activity list. If an agent can cover all the activities of tourist j and the cost is minimum, then that agent will win the service for the tourist j . If there is no single agent that can cover the tourist j 's activity list, then there will be a consortium of agents for the service of tourist j . For each activity on the list of desired activities of the tourist, tourist agents will compete in the auction. A consortium will be formed for this given tourist activity list. This consortium will cover all the activities in the tourist's activity list at a minimum price.

Matching Set

1. Tourist Set $T = \{t_1, \dots, t_m\}$

TA_j : tourist (t_j) j^{th} Activity list. $TA_j = \{a_1, \dots, a_n\}$

2. Agent Set $A = \{a_1, \dots, a_n\}$

It is the agent i 's activity list $SA_i = \{(a_{in}, p_{in}), \dots, (a_{im}, p_{im})\}$ where a_{in} stands for the agent i 's activity n and its price is p_n .

3. Consortium $\bigcup_{k \in A} C_k = TA_j$ (set cover)

The consortium is a union of a minimum set of agents' selected activity list and it covers TA_j

where

$C_K = \{(a_{k,p}, p_{k,p}) \mid k \in A, a_{k,p} \in TA_k, p_{k,p} = \min(p_m, p) \text{ for all } m \in A \text{ and } a_{k,p} \in TA_j\}$

Note: $a_{k,p}$ represents agent k 's activity list p and $p_{k,p}$ represents the price of agent k 's activity p .

C_K is a consortium of activities of service a_k providers.

Example:

There is a tourist John, and he has a list of activities,

TA "John" = $\{a_1, a_2, a_3, a_4, a_5\}$

There are three agents:

$A = \{S_1, S_2, S_3\}$

Each agent has a service list with prices:

$SA_1 = \{(a_1, 10), (a_2, 20)\}$, $SA_2 = \{(a_1, 20), (a_4, 10), (a_3, 10)\}$, $SA_3 = \{(a_4, 20), (a_5, 10)\}$

The service consortium for John can be selected as follows:

$C_{\text{John}} = \{(a_{11}, 10), (a_{12}, 20), (a_{23}, 10), (a_{24}, 10), (a_{35}, 10)\}$

Each activity is selected at the minimum price from different agents, and this list will cover John's activity list.

4 Implementation

4.1 Java Model

This paper uses Java as a modeling tool and a heuristic algorithm for the tourism service auction market. Figure 2 shows the use of Java for the proposed auction market with agents, and the consortium is coded by using Java programming. Moreover, Fig. 3 displays a sample run of the example in our math model.

4.2 Implementation Results

There were a total of 20 runs with randomized data. An example of creating tourist agents and their services is demonstrated in Fig. 4, and tourists' wanted activities are listed in Fig. 5. In these examples, the proposed heuristic algorithm successfully made consortia for the tourists' activity requirement list. As a sample, results for two tourists' activities lists and service matchings are displayed in Fig. 6.

5 Conclusion

In this paper, a model for an electronic tourism service auction market for Saudi Arabia has been proposed, as this will lead to a more efficient tourism support mechanism in Saudi Arabia. In this electronic tourism auction market, a mathematical model is proposed and implemented with a heuristic algorithm in Java. Based upon random tourist and agent service auction trading, this model was

Fig. 2 Heuristic algorithm for the tourism service auction market

```
function match (Tourst_Activity,List of activities of agents)
{
  foreach Agent
    look for activities requested by a tourist
    If current agent can cover all the activities
      Form the tourist with minimum
      Return the Agent and its activities
    Else
      Apply consortium to find all sets of Tourist Activities
      (TA) with price_check
      function price_check
        select an activity of an agent
        with minimum price
        Return the selected Agent, its activities with price
  }
```

There are three agents {1,2,3}

Each agent has a service list with prices:

Agent 1 has Activities:
Activity no. 1 with price: 10,
Activity no. 2 with price: 20,

Agent 2 has Activities:
Activity no. 1 with price: 20,
Activity no. 4 with price: 10,
Activity no. 3 with price: 10,

Agent 3 has Activities:
Activity no. 4 with price: 20,
Activity no. 5 with price: 10,

There is a tourist John and he has a list of activities:
John wants Activities: 1, 2, 3, 4, 5,

The services consortium from different agents for John can be selected as follows:

John Activity No 1 match Agent1
John Activity No 2 match Agent1
John Activity No 3 match Agent2
John Activity No 4 match Agent2
John Activity No 5 match Agent3

Each activity is selected at the minimum price from different agents, and this list will cover John's activity list and the package total price for John is: 60

Fig. 3 A sample run of the example in Sect. 3.1

Fig. 4 Agents, their activities, and prices

The Agents (service provider) are:

Agent 1 has Activities:
Activity no. 19 with price: 44,
Activity no. 14 with price: 54,
Activity no. 18 with price: 48,
Activity no. 5 with price: 57,
Activity no. 9 with price: 48,

Agent 2 has Activities:
Activity no. 18 with price: 42,
Activity no. 19 with price: 52,
Activity no. 1 with price: 49,
Activity no. 2 with price: 58,
Activity no. 12 with price: 46,

Agent 3 has Activities:
Activity no. 20 with price: 50,
Activity no. 8 with price: 53,
Activity no. 16 with price: 49,
Activity no. 13 with price: 46,
Activity no. 12 with price: 51,

executed successfully and demonstrated the performance of basic functionalities of the tourism service market. The proposed model selects activities with minimum prices for a tourist's list of desired activities through agent consortia. The results

Fig. 5 A sample case of ten tourists’ requested activities

The Tourists desired activities are:
 Tourist 1 wants Activities: 5, 7, 19, 8, 6, 1,
 Tourist 2 wants Activities: 16, 8, 9, 20, 10, 2,
 Tourist 3 wants Activities: 2, 11, 7, 8, 20, 18,
 Tourist 4 wants Activities: 5, 4, 8, 16, 18, 7,
 Tourist 5 wants Activities: 7, 4, 14, 15, 13, 1,
 Tourist 6 wants Activities: 16, 1, 7, 8, 18, 19,
 Tourist 7 wants Activities: 11, 4, 19, 20, 1, 12,
 Tourist 8 wants Activities: 10, 14, 3, 17, 5, 15,
 Tourist 9 wants Activities: 14, 4, 16, 18, 3, 8,
 Tourist 10 wants Activities: 12, 16, 6, 17, 13, 14,

The results of desired activities for tourists with minimum prices for a tourist's list through agent consortium:

Tourist 1 :
 Activity number 5 Match Agent 11 and the price is: 40
 Activity number 7 Match Agent 9 and the price is: 43
 Activity number 19 Match Agent 1 and the price is: 44
 Activity number 8 Match Agent 10 and the price is: 42
 Activity number 6 Match Agent 9 and the price is: 41
 Activity number 1 Match Agent 17 and the price is: 42

The package total price for Tourist 1 is: 252

Tourist 2 :
 Activity number 16 Match Agent 14 and the price is: 43
 Activity number 8 Match Agent 10 and the price is: 42
 Activity number 9 Match Agent 16 and the price is: 40
 Activity number 20 Match Agent 12 and the price is: 46
 Activity number 10 Match Agent 8 and the price is: 43
 Activity number 2 Match Agent 13 and the price is: 40

The package total price for Tourist 2 is: 254

Fig. 6 A Sample results for two tourists’ activity lists

of the model also demonstrate the usefulness of agent consortia for fulfilling the tourist’s list of desired activities at a minimum price from various service providers. As demands for tourism services in Saudi Arabia increase under the 2030 Vision, an effective electronic market model will contribute to its cause. Overall, the proposed model demonstrates a feasible solution for the tourism services in Saudi Arabia, which utilizes an electronic auction market with consortia.

References

1. Centre, UNESCO World Heritage. World Heritage List. UNESCO World Heritage Centre. Accessed 24 Feb 2020. <https://whc.unesco.org/en/list/>
2. National Transformation Program. National Transformation Program | Saudi Vision 2030. Accessed 24 Feb 2020. <https://vision2030.gov.sa/en/programs/NTP>
3. About Us. Royal Commission For Alula, Home. Accessed 24 Feb 2020. <https://www.rcu.gov.sa/en/>
4. A. Farzammia, S.M.R. Nasserzadeh, S. Nalchigar. Which Internet Marketing Mix’s Has More Effect on the Passenger’s Decision for Choosing Their Travel Agency in Iran? In INC, IMS and IDC, 2009. NCM’09. Fifth International Joint Conference on (pp. 1087–1092). IEEE (2009, August)

5. P. Cipparrone. Introduction to the Internet. San Diego County Library (2006). Retrieved 1 Dec 2011 from http://sdcl.org/PDF/gateway_introduction-to-internet.pdf
6. Quirk. (2006). What is eMarketing? Retrieved 24 Feb 2020 from <http://www.quirk.biz/resources/88/What-is-eMarketing-and-how-is-it-better-than-traditional-marketing>
7. J. Kim. B2B E-commerce in the Printing Industry Part 1. 5th Annual Meeting of the Forum of Asian Graphic Arts Technology (FAGAT) in Manila (2000). Retrieved February 24, 2020 from http://www.jagat.or.jp/story_memo_view.asp?StoryID=3711
8. D. Chaffey, R. Mayer, K. Johnston, F. Ellis-Chadwick, *Internet Marketing: Strategy, Implementation and Practice* (Pearson Education, NYC, NY, USA, 2002)
9. M. Popović. B2B e-Marketplaces (2002). Available in: <http://www.teleactivities.net>. Retrieved 24 Feb 2020 from <http://www.teleorg.org/e-commerce/studies/B2Bemarketplaces.doc>
10. J.R. Milton, International consortia: Definition, purpose and the consortium agreement. *Fordham Int. Law Forum* **3**, 121–140 (1979-1980)
11. L. Cassivi, A.L. Saives, E. Labzagui, P. Hadaya. Developing a Knowledge Sharing Platform: The Case of a Bio-Industry Research Consortium. In *Information, Process, and Knowledge Management, 2009. eKNOW'09. International Conference on* (2009, February), pp. 153–158. IEEE
12. M. Ivanovic, et al. Establishing A Consortium – Way for Successful Implementation of Investments Projects - An Example of The Infrastructural Project ‘Slavonian Networks’. **3**, 28–36 (2014)
13. Local t-gov. (2007). Buying consortia. Retrieved 24 Feb 2020 from <http://www.localt.gov.uk/webfiles/NePP/Guidance/3.0%20Collaboration/3.1.2.pdf>
14. C. Soh, M.L. Markus, B2B E-market places–interconnection effects, strategic positioning, and performance. *Systemes d’information et Management* **1**(7), 77–103 (2002)
15. D. Loader, *Clearing and Settlement of Derivatives. Clearing, Settlement and Custody* (Butterworth-Heinemann, Oxford, 2014), pp. 55–95. <https://doi.org/10.1016/b978-0-08-098333-2.00005-0>
16. D.M. Chance, R.E. Brooks, *An Introduction to Derivatives and Risk Management* (South-Western Cengage Learning, México, 2010)
17. E. Paul. Black, “set cover”, in *Dictionary of Algorithms and Data Structures* [online], Paul E. Black, ed. 1 March 2020. (accessed Today) Available from: <https://www.nist.gov/dads/HTML/setcover.html>
18. W. Tian, *On The Duality Feature of P-Class Problems and NP Complete Problems* (Computer Science & Information Technology, 2018). <https://doi.org/10.5121/csit.2018.80303>
19. V.V. Vazirani, *Approximation Algorithms* (Springer, Berlin, 2003)
20. N. Bilal, P. Galinier, F. Guibault, A new formulation of the set covering problem for metaheuristic approaches. *ISRN Oper Res* **2013**, 1–10 (2013). <https://doi.org/10.1155/2013/203032>
21. B. Yelbay, Ş.I. Birbil, K. Bülbül, The set covering problem revisited: An empirical study of the value of dual information. *J Ind Manag Optim* **11**(2), 575–594 (2015). <https://doi.org/10.3934/jimo.2015.11.575>
22. A. Caprara, M. Fischetti, P. Toth, Algorithms for the set covering problem. *Ann. Oper. Res.* **98**, 353–371 (2000)

The Use of Crowdsourcing as a Business Strategy



Hodaka Nakanishi and Yuko Syozugawa

1 Introduction

Recently, the number of companies which utilize crowdsourcing is increasing. For companies, crowdsourcing is a new business management method of procuring resources from the “crowd” via the Internet, rather than the traditional method of procuring required resources from other companies. The use of crowdsourcing is increasing in Japan as well. In 2013, Japan Small and Medium Enterprise Agency conducted a “survey on the actual usage of crowdsourcing in Japan.” Based on the survey, the White Paper on Small and Medium Enterprises 2014 stated that the market size of crowdsourcing would increase six times in 4 years [1]. Considering that it is difficult for small and medium-sized enterprises (SMEs) to secure human resources, the Small and Medium Enterprises Agency carries out the dissemination and enlightenment of crowdsourcing as a means of SMEs’ management support such as disclosure of know-how and introduction of use cases for facilitating the use of crowdsourcing.

The concept of crowdsourcing has been found since around 2000, like Amazon’s Mechanical Turk and iStock, but the term “crowdsourcing” was first used by Howe in 2006. He defined the term as “crowdsourcing represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call [2].”

H. Nakanishi (✉)
Teikyo University, Tokyo, Japan
e-mail: nakanishi@med.teikyo-u.ac.jp

Y. Syozugawa
Tokoha University, Shizuoka, Japan
e-mail: shozugaway@sz.tokoha-u.ac.jp

In this paper, crowdsourcing is defined as the act of outsourcing to an unspecified person such as freelance using the Internet. This definition is in line with the definition in the White Paper on Small and Medium Enterprises in Japan 2014: “Crowdsourcing is a mechanism for contacting large numbers of individuals and enterprises through the Internet in order to procure needed human resources (p.373) [1].” Although the use of the Internet is not essential for crowdsourcing originally, the definition in this paper adds the use of the Internet to the often-quoted definition of Afuah and Tucci, “Crowdsourcing is the act of outsourcing a task to a ‘crowd,’ rather than to a designated ‘agent’ (an organization, informal or formal team, or individual), such as a contractor, in the form of an open call (p.355) [4], because the procurement method using the Internet has spread recently [3].”

Crowdsourcing is expected to have the effects of contributing to the innovative activities in companies such as problem-solving and development of new product [4, 5], but at the same time, the limits of crowdsourcing have also been pointed out [6, 7]. Although the evaluation of crowdsourcing varies, few studies have examined how companies regard crowdsourcing as a corporate strategy. Especially in Japan, most of the crowdsourcing surveys are related to crowdsourcees (workers of crowdsourcing), and there are only a few surveys on crowdsourcers (the companies that use crowdsourcing) [1, 8]. Even in the surveys on crowdsourcers, the relation between crowdsourcing and corporate strategy has not been analyzed. The survey conducted by the small and medium enterprise agency is based on the answers of the registrants of the crowdsourcing site who have actually ordered work on the crowdsourcing site and reached the contract. As many respondents are self-employed (35% of the total) and few respondents working for companies [1], the results of the survey do not reflect the situation of the company.

The purpose of this research is to clarify how crowdsourcing can contribute to the management of a company by studying the reason why companies use crowdsourcing and its effect together with the characteristics of the company such as scale, management policy, and so on by the questionnaire.

The next chapter organizes the research of the effects of crowdsourcing on corporate strategy and sets the taxonomy of the purpose and effect of crowdsourcing. In Chap. 3, we will explain the outline of the methods, targets, etc. of the survey of crowdsourcing utilization situation in companies. Chapter 4 shows the results of the survey and analyzes them according to the taxonomy set in Chap. 2. Finally, Chap. 5 discusses the results.

2 Taxonomy of Crowdsourcing for Business Strategy

There are some studies on the purpose of crowdsourcing by companies or organizations. For example, Howe presents four types of crowdsourcing: the professional, the packager, the tinkerer, and the masses [9]. As for the professional, Howe shows the example of the National Health Museum which used iStockphoto, a crowdsourcing site, to reduce costs. For the packager, he shows an example of a

Table 1 Taxonomy of crowdsourcing for business strategies

4 Business strategy elements and 19 items of the questionnaire	Items in [1]
Procurement of management resources (MR)	
You can make up for the lack of internal management resources	x
You can substitute the employment of workers who is in charge of the task	x
You can identify contractors' skills	x
You can understand your own strengths and weaknesses	
You can publicize your company	x
Improvement of productivity (IP)	
You can speed up the business	x
You can choose from a large number of contractors	x
You can select a contractor after checking the results	x
You can concentrate management resources on business areas where you focus on	
You can receive high-quality outcome	x
Costs reduction (CR)	
You can reduce costs to secure necessary human resources or systems by yourself	x
You can reduce total working hours	
You can reduce employee's workload	
You can review existing businesses to improve efficiency and to reduce costs	
Responding to fluctuations (RF)	
You can respond to an increase of orders	x
You can respond to an increase of business partners	x
You can order only when you need it	x
You can respond to changes in the amount of work	x
Others	x

viewer-posted TV program using crowdsourcing to collect information (contents) from a crowd. An example of the tinkerer is InnoCentive, where companies use crowdsourcing to obtain solutions from the outside. An example of the masses is Mechanical Turk, which procures the tasks that companies need from a large number of external human resources (crowd) to carry out their business. Bauer and Gegenhuber explains that there are four types of value that crowdsourcing creates. Those are the possibilities of access to (1) creative expertise and (2) critical items and the increase of (3) execution capacity and (4) bargaining power (p.666) [10]. Erickson (2012) identified productivity, innovation, knowledge capture, and marketing/branding as the four common uses of the crowd (p.93) [11]. In the White Paper on Small and Medium Enterprises 2014, Japan Small and Medium Enterprise Agency conducts a questionnaire survey which asks the reasons for using crowdsourcing from 14 items [1]. The items are listed in Table 1. In the survey, more than half of the respondents chose “You can order only when you need it” (64.9%), “You can make up for the lack of internal management resources such as technology and human resources” (57.3%), “You can receive high-quality outcome” (54.5%), and “You can speed up the business” (50.0%) [1].

In this paper, in order to understand the effect of the use of crowdsourcing on business strategies, 5 items were added to the 14 items used in the white paper, and a total of 19 items were set (see Table 1). Based on previous studies, the relationship between crowdsourcing and corporate management can be categorized into four business strategy elements: “procurement of management resources (MR),” “improvement of productivity (IP),” “cost reduction (CR),” and “responding to fluctuation (RF).” We make up a taxonomy that applies the 19 items to the 4 business strategy elements (Table 1).

3 Procurement of Management Resources (MR)

Procurement of resources necessary for the management such as human resources, technology, and knowledge from the cloud is recognized as a major role of crowdsourcing. Problem-solving and innovation creation are often considered to be effects of crowdsourcing, and they can be considered as a final goal of management resource procurement. When companies are trying to solve their problems, crowdsourcing is used to procure “knowledge” when there is no knowledge for the solution within the company [4]. In addition, crowdsourcing is often used when companies try to create innovation. The tinkerer [9], like InnoCentive, procures knowledge from the cloud and is used to generate innovation [12]. Poetz and Schreier also assert the effectiveness of crowdsourcing as a means of gathering ideas when developing new products [5]. In the questionnaire, the following five items were set as the items indicating “procurement of management resources.”

- You can make up for the lack of internal management resources (such as technology, human resources, know-how).
- You can substitute the employment of workers who is in charge of the task.
- You can identify contractors’ skills (for future direct employment or continuous ordering).
- You can understand the strengths and weaknesses of the company.
- You can publicize your company.

4 Improvement of Productivity (IP)

Crowdsourcing is often used to improve productivity. Improving productivity includes improving operational efficiency and innovation of production processes. Erickson points out that the improvement of productivity is one of the results obtained by crowdsourcing based on his interviews with people who work for large companies [10]. Bauer and Gegenhuber also state that crowdsourcing produces value qualitatively and quantitatively [3]. Thus, improving productivity is one of

the important expected effects of crowdsourcing. In the questionnaire, the following five items were set as the items indicating “improvement of productivity.”

- You can speed up the business.
- You can choose from a large number of contractors.
- You can select a contractor after checking the results.
- You can concentrate management resources on business areas where you focus on.
- You can receive high-quality outcome.

5 Costs Reduction (CR)

Cost reductions, including reductions in raw material costs and labor costs, are also pointed out as crowdsourcing effects. Howe shows an example of purchasing a photo at a low price using a crowdsourcing site, iStockphoto, in the explanation of the professional, one of his crowdsourcing categories [9]. Ericson points out that cost reduction factors such as financial motivation and value capture are organizational characteristics of crowdsourcing [10]. Schenk and Guittard also show that one of the major advantages of crowdsourcing is low cost [12]. In the questionnaire, the following four items were set as the items indicating “cost reduction.”

- You can reduce costs to secure necessary human resources or systems by yourself.
- You can reduce total working hours.
- You can reduce employee’s workload.
- You can review existing businesses to improve efficiency and to reduce costs

6 Responding to Fluctuations (RF)

Bauer and Gegenhuber (2015) argue that there are four types of value that crowdsourcing produces, and one of them is the ability to respond to increasing demand [3]. Responding to fluctuations in demand is also considered an effect of crowdsourcing. In the questionnaire, the following four items were set as the items that indicate “responding to fluctuations.”

- You can respond to an increase of orders.
- You can respond to an increase of business partners.
- You can order only when you need it.
- You can respond to changes in the amount of work.

These four elements are not independent of each other, and multiple elements may relate to the same purpose or effect at the same time. However, all of these

elements can be considered as the role of crowdsourcing. In this paper, we analyze the purpose and the effect of crowdsourcing for companies in Japan with this taxonomy.

7 Outline of the Survey

This survey is an internet survey using a panel consisting of monitors registered in “NTTCom Research.” The survey was outsourced to NTTCom Online Marketing Solutions Corporation. The panel monitors are men and women aged 20 to 69 years old living in Japan, with a total of 2.17 million members. At first, we conducted the screening survey to the panel monitors which ask, “At your workplace, do you use crowdsourcing which outsources business to unspecified people (such as freelance)?” A total of 1010 respondents who answered that they had experience in ordering with crowdsourcing were surveyed. The survey was conducted for 2 days from September 25 to 26, 2019.

In order to clarify the attributes of the respondents, we asked about occupations, industry classification of the company, company sizes, job titles, and the contents of ordering tasks with crowdsourcing. About 78% of the respondents are full-time employees or executives working for companies, and only 13% are self-employed. In this survey, we mainly analyzed the responses of company employees and executives to understand the situation of companies. The data of self-employed were referred for the comparison as necessary.

A question might arise whether the company’s policy can be understood by asking questions to employees. However, in the preliminary interviews with those who had experience using crowdsourcing in the company, there was information that the use of crowdsourcing was decided not by the company, but by the unit of division or individual. That is, information that reflects the use condition of crowdsourcing in a company cannot always be obtained even if a questionnaire survey is conducted on a company-by-company basis. In fact, as a result of the survey, only 31.9% answered that the company decided to use crowdsourcing, while 45.6% of the respondents answered that they decided to use crowdsourcing on an individual department basis (Table 2). In addition, it is often difficult to obtain a sufficient number of responses in a questionnaire to companies. We therefore conducted a survey targeting individuals.

The White Paper on Small and Medium Enterprises [1] is also investigating the reasons for introducing crowdsourcing to individuals as well. However, the White Paper on Small and Medium Enterprises does not investigate the attributes of the companies to which the individual belongs, so the situation of crowdsourcing depending on the characteristics of the company is unknown. In this study, we investigate the reasons and effects of using crowdsourcing and get the information of the companies to which the respondents belong simultaneously.

The industries to which the most respondents belong are manufacturing (25.5%), followed by education, medical, miscellaneous services (19.7%), computer and

Table 2 Decision level for the use of crowdsourcing

	%
Company level	31.9
Division or department level	34.1
Personal level	11.5
Depends on the matter	15.4
Unknown	6.8
Others	0.3

Table 3 The industry the respondents of the questionnaire belong to

Industry	%
Manufacturing	25.5
Wholesale and retail trade	8.6
Finance and insurance	5.7
Communications	6.5
Transport, construction, and real estate	11.4
Computer and information services	14.2
Education, medical, miscellaneous services	19.7
Others	8.3

Table 4 Task of crowdsourcing

Tasks	%
IT, apps, and programs related to business	25.4
Human resource operation, education, seminar lecturer	24.2
Data input, data processing, data classification	20.7
Management planning, management strategy, M&A	18.5
Financial accounting	14.0
Domestic business, sales, marketing	13.1
Design such as logo creation	12.7
Guidance of management and service improvement	12.5
Writing, naming	11.8
Survey (web, overseas market, questionnaire, etc.)	11.7
Video/multimedia/advertising	11.1
Product development and business idea creation	8.1
Translation	7.5
Transcription	5.2
Overseas business	4.4
Others	2.0

information services (14.2%), and transport, construction, and real estate (11.4%) (Table 3). This figure shows that the percentage of manufacturing industry is higher, and the percentage of retail commerce is lower compared to the percentage of employees by industry in Japan as a whole.

Looking at the contents of the ordering tasks of the respondents, “IT, apps, and programs related to business” was the most common with 25.4% of the respondents, followed by “human resource operation, education, seminar lecturer” (24.2%) and “data input, data processing, data classification” (20.7%) (Table 4).

Table 5 Number of respondents by company size

	Number	%
Large companies	626	62.0
Small- to medium-sized companies	282	27.9
Self-employed	102	10.1
Total	1010	100.0

As for the size of the companies to which the respondents belong, 62% of respondents work for large companies, 28% work for SMEs, and 10% are self-employed (Table 5). Many of the respondents are employees of large companies. Thus, it can be considered that the results of this survey reflect the condition of crowdsourcing of companies as a whole. Here, a large company refers to 300 or more employees in the manufacturing industry and 100 or more employees in the nonmanufacturing industry. SMEs are 2 to 299 employees in the manufacturing industry and 2 to 99 employees in the nonmanufacturing industry. One employee is classified as self-employed. This definition of SMEs complies with the definition of those by the Small and Medium Enterprise Agency. Erickson indicates that “small- to medium-sized organizations were unlikely to be leveraging crowdsourcing practices due to perceived risks, costs, and lack of available resources (p.93)” and chooses mature companies with a minimum of 500 employees as the case study sites [10]. In our survey too, there are many responses from employees of large companies. Crowdsourcing may tend to be used by large companies.

Regarding the survey of 2014 White Paper on Small and Medium Enterprises, whose data is based on a survey conducted by W’sSTAFF [13], 67.0% of the respondents of employees are self-employed, 27.6% belong to companies of 100 or less employees, which is almost equivalent to SME, and 5.3% belong to the companies with more than 101 employees, which is almost equivalent to a large company. Therefore, the SME White Paper mainly reflects the situation of self-employed or small businesses. On the other hand, our survey mainly reflects the condition of the utilization of crowdsourcing in companies including large companies and SMEs.

In this survey, we asked for which business type the crowdsourcing was introduced, a main business, a non-main business, or a new business for the company. This is a question to investigate whether a company uses crowdsourcing from the viewpoint of business strategy.

Of the 908 responses from companies that use crowdsourcing, 47.5% use crowdsourcing for main businesses, 41.2% for non-main businesses, and 24.3% for new businesses (Table 6). It should be noted that the difference in the ratio of crowdsourcing by business type reflects the difference in the number of companies working on each business and does not indicate only the difference in the ratio of crowdsourcing introduction by business type. Also, when crowdsourcing is used in multiple types of business, duplicate answers are allowed, so the total number of companies for each item does not match the total number of companies (908 companies).

Table 6 The use of crowdsourcing by business type (multiple answers)

	Number of companies	Percentage of companies
Main business	431	47.5%
Non-main business	374	41.2%
New business	221	24.3%
Total	908	100.0%

Table 7 Objectives of crowdsourcing before and after the introduction

	MR	IP	CR	RF
Reasons (before)	51.2%	51.7%	34.0%	44.1%
Effects (after)	52.3%	50.4%	35.8%	45.7%

8 Result

8.1 Reasons and Effects in Total

The 19 items of the reasons and effects of the use of crowdsourcing by companies are classified into four management strategy elements (see Table 1). If one or more items in each element are checked, the element is counted as checked. The number of checked items in an element is not considered in the analysis because the number of items in one element is not the same among them. The reason for using crowdsourcing is considered to be the expectation for crowdsourcing, and the effect is the ex-post evaluation.

More than half of the companies indicate improvement of productivity (51.7%) and procurement of management resources (51.2%) as the reason for introducing crowdsourcing (Table 7). With regard to the effects brought by crowdsourcing, more than half of the companies could raise productivity (50.4%) and could procure the management resources (52.3%). These numbers indicate that crowdsourcing is introduced with the expectation of productivity and procurement and that they are actually effective. On the other hand, only 34.0% of companies are targeting cost reduction for crowdsourcing, and 35.8% recognize the effects. It means companies often use crowdsourcing to improve productivity and procure management resources, while they give less attention to cost reduction effects. As there is no significant difference between the ratio of reason and that of effect with the two-proportion z-test results, companies are supposed to get the results as expected in the introduction of crowdsourcing.

On the item basis for the reasons and results of using crowdsourcing, many companies mention “You can make up for the lack of internal management resources” (MR), followed by “You can receive high-quality outcome” (IP) (Table 8).

Table 8 Top five reasons and effects of crowdsourcing for companies

Rank	Reasons	%	Effects	%
1	You can make up for the lack of internal management resources (MR)	37.0	You can make up for the lack of internal management resources (MR)	37.2
2	You can receive high-quality outcome (IP)	32.0	You can receive high-quality outcome (IP)	31.1
3	You can respond to an increase of orders (RF)	24.7	You can speed up the business (IP)	21.7
4	You can speed up the business (IP)	23.5	You can respond to an increase of orders (RF)	21.3
5	You can reduce costs to secure necessary human resources or systems by yourself (CR)	20.2	You can reduce costs to secure necessary human resources or systems by yourself (CR)	20.5

Table 9 Company size and the reason of crowdsourcing

	MR	IP	CR	RF
SME	50.4%	45.4%	33.7%	45.7%
Large companies	51.6%	54.5%	34.2%	43.3%
Company total	51.2%	51.7%	34.0%	44.1%
(Personal business)	(61.8%)	(42.2%)	(28.4%)	(32.4%)

8.2 Reasons and Effects by Company Size

The reasons for the introduction of crowdsourcing are compared by company size (Table 9). As a result of the two-proportion z-test to compare the ratios of the selection items of the reason for the introduction of crowdsourcing between large companies and SMEs, large companies are more likely to improve productivity than SMEs (large companies: 54.5%, SMEs: 45.4%, $p < 0.05$). On the item basis, the ratio of companies which point out “You can receive high-quality outcome” was particularly high in large companies (large companies: 34.2%, SMEs: 27.3%, $p < 0.05$). The result indicates that larger companies tend to aim for the improvement of productivity by crowdsourcing. The ratio of companies that procure management resources is the highest among the four factors. The procurement of management resources is considered to be important for both large companies and SMEs as there is no significant difference between the ratio of large companies and that of SMEs. Only one-third of the companies introduce crowdsourcing for cost reduction reasons. There are not many large companies and SMEs that introduce crowdsourcing reasoning cost reduction. However, on the item basis, the ratio of “You can reduce total working hours” is significantly higher in large companies (large companies: 10.5%, SMEs: 6.0%, $p < 0.05$). Large companies may use crowdsourcing for work style reform which is an important issue in Japan.

The percentage of self-employed who raise management resources as a reason for introducing crowdsourcing is 61.8%, which is significantly higher ($p < 0.05$)

than the average of enterprises (51.2%). This indicates that self-employed persons are attracted to crowdsourcing because they lack management resources.

For the results (effects) of the introduction of crowdsourcing, many large companies and SMEs point out “procurement of management resources” and “improvement of productivity,” but there is no big difference between the two factors (Table 10). For both factors, large companies show a significantly higher ratio of 8–9% points than SMEs. Large companies recognize the effects of crowdsourcing more than SMEs do. The same tendency can be seen in the ratios of items for large companies and SMEs, but the answer “You can order only when you need it,” which is an item of responding to fluctuations, is significantly higher in SMEs (large companies: 14.2%, SMEs: 21.6%, $p < 0.01$).

8.3 Reasons and Effects by Business Type

The reasons for using crowdsourcing are compared by business type, i.e., main businesses, non-main businesses, and new businesses (Table 11). In main businesses, procurement of management resources is the most common answer, while improvement of productivity is the most common answer in non-main businesses and new businesses. In new businesses, the proportion of companies that use crowdsourcing to improve productivity and reduce costs is significantly higher than in main businesses.

Some items are characterized by business type. “You can make up for the lack of internal management resources” (MR) is pointed out by many companies in any business type. The ratio of “You can reduce costs to secure necessary human resources or systems by yourself” (CR) is significantly higher in new business (29.4%, $p < 0.01$) and non-main business (26.2%, $p < 0.05$) than in main business (19.5%). The ratios of some other items such as “You can speed up the business” (IP) (33.0%, main: 22.7%, $p < 0.01$) and “You can order only when you need it”

Table 10 Company size and effects of crowdsourcing

	MR	IP	CR	RF
SME	46.8%	44.3%	33.3%	45.7%
Large companies	54.8%	53.2%	36.9%	45.7%
Company total	52.3%	50.4%	35.8%	45.7%
(Personal business)	(48.0%)	(42.2%)	(28.4%)	(32.4%)

Table 11 Business type and the reason for the introduction of crowdsourcing

	MR	IP	CR	RF
Main business	56.1%	52.7%	33.2%	48.5%
Non-main business	50.5%	53.2%	39.3%	44.1%
New business	59.3%	62.9%	42.5%	55.7%
Total	51.2%	51.7%	34.0%	44.1%

Table 12 Business type and effects of the introduction of crowdsourcing

	MR	IP	CR	RF
Main business	57.8%	56.1%	33.6%	51.7%
Non-main business	51.9%	48.1%	43.0%	46.5%
New business	62.9%	63.3%	43.9%	54.3%
Total	52.3%	50.4%	35.8%	45.7%

(RF) (22.2%, main: 13.5%, $p < 0.01$) are also significantly higher in new business than in main business.

This indicates that companies are trying to efficiently procure management resources by incorporating crowdsourcing into their business processes when implementing main businesses, whereas they are actively using crowdsourcing to improve productivity and reduce costs for new businesses and non-main businesses and are trying to implement their businesses in a way that does not place a burden on their main businesses.

The outcomes of the introduction of crowdsourcing tend to be perceived at a higher ratio for any one element in new business than in the main business. In particular, the cost reduction effects are significantly higher in both new business (43.9%, $p < 0.01$) and non-main business (43.0%, $p < 0.01$) than in the main business (33.6%) (Table 12).

On the item basis, “You can make up for the lack of internal management resources” (MR) is the most perceived effect of crowdsourcing in any type of business. In addition, as in the case of the reason for use, the ratio of the item “You can reduce costs to secure necessary human resources or systems by yourself” (CR) in the new business and non-main business is significantly higher than main business. For many items such as “You can speed up the business” (IP), the ratio of items of crowdsourcing results is significantly higher in new business than in main business.

These findings indicate that results of crowdsourcing introduction are recognized especially in new businesses.

9 Conclusion

An analysis of the reasons for using crowdsourcing by the four categories (i.e., procurement of management resources, improvement of productivity, cost reduction, and responding to fluctuations) reveals how companies are using crowdsourcing as a management strategy.

More than half of the companies point out that improvement of productivity and procurement of management resources are their reasons for introducing crowdsourcing, and they recognize these effects as well. This indicates that crowdsourcing has been positioned in the company’s management strategy. At the same time, only about one-third of companies have adopted crowdsourcing for a cost reduction tool,

indicating that crowdsourcing is not simply introduced to make use of cheaper labor force in crowds.

By company size, large companies are more likely to adopt crowdsourcing to improve productivity. In addition, a higher ratio of large companies is using crowdsourcing to reduce working hours than SMEs. Crowdsourcing contributes to the working style reforms, which is an important policy for the Japanese government, especially in large companies.

By business type (main business, non-main business, and new business), more companies in the new business category point out productivity improvement and cost reduction as reasons for using crowdsourcing than in the main business category. This may indicate that many companies are using crowdsourcing to tackle new businesses in a way that does not put a burden on their main businesses.

The results of this study show that crowdsourcing is used as an innovative tool for management and as part of management strategy, especially by large companies. As the development of innovative products and services becomes more difficult in a mature society, it is expected that crowdsourcing will play an important role in the future growth of companies, including the development of new business fields. As for a further study of crowdsourcing as a business strategy, we would like to analyze from different perspectives such as long-term trends and differences by industry.

Acknowledgments This work was supported by Grants-in-Aid for Scientific Research, Grant Number 18 K01764, awarded by the Japan Society for the Promotion of Science.

References

1. The Small and Medium Enterprise Agency. *2014 White Paper on Small and Medium Enterprises in Japan* (2014)
2. J. Howe. *Crowdsourcing: A Definition*, June 02, 2006 (2006). http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html. Retrieved on 4 Feb 2018
3. R.M. Bauer, T. Gegenhuber. Crowdsourcing: Global search and the twisted roles of consumers and producers. *Organization* **22**(5), 661–681 (2015)
4. A. Afuah, C.L. Tucci, Crowdsourcing as a solution to distant search. *Acad. Manag. Rev.* **37**(3), 355–375 (2012)
5. M.K. Poetz, M. Schreier, The value of crowdsourcing: Can users really compete with professionals in generating new product ideas? *J. Product Innov. Manag.* **29**(2), 245–256 (2012)
6. J. Bloodgood, Crowdsourcing: Useful for problem solving, but what about value capture? *Acad. Manag. Rev.* **38**(3), 455–457 (2013)
7. I. Christensen, C. Karlsson, Open innovation and the effects of crowdsourcing in a pharma ecosystem. *J. Innov. Knowl.* **4**, 240–247 (2019)
8. ULULU. Attitude survey of the use of crowdsourcing by small and medium enterprises, March 5, 2015 (in Japanese) (2015). https://www.uluru.biz/wp-content/uploads/2015/03/150305_shufti.pdf. Retrieved on 13 Apr 2020
9. J. Howe. The rise of crowdsourcing, *Wired Magazine* (2006) 14.06, June 2006
10. L.B. Erickson, *Leveraging the Crowd as a Source of Innovation: Does Crowdsourcing Represent a New Model for Product and Service Innovation?*, SIGMIS-CPR '12, May 31–June 2 (Milwaukee, Wisconsin, 2012)

11. T.C. Monteiro, A.L. Zambalde, P.H. de Souza Bermejo. Crowdsourcing aimed at value innovation, *Proceedings of the 52nd Hawaii International Conference on System Sciences* (2019)
12. E. Schenk, C. Guittard, Towards a characterization of crowdsourcing practices. *J. Innov. Econ. Manag.* **7**, 93–107 (2011)
13. Y's STAFF. Nihon no crowdsourcing kankyo ni kansuru tyosa houkokusyo (Survey of Crowdsourcing Environment in Japan) (in Japanese) (2014)

Index

A

- ABET accreditation, 85, 87, 90, 747, 749, 752
- Abstract Conformance Test Suite (A.C.T.S.), 550, 551, 553
- Academia, 179, 265, 266, 494, 741, 750
- Academic CS, 180–189, 192
- Academic knowledge, 239, 240, 245
- Accelerometers, 109, 110, 114
- Active learning, 8, 35, 250
 - basic control course, 251
 - in computer science, 145
 - and CT, 218–221
 - definition, 144
 - MIPU (*see* Minimum pumping length (MIPU))
 - OOD, 147
 - pumping lemma property, 143
 - STEM education, 144
- Actuators, 200, 201, 205, 207
- Adaptation
 - agile ITSM framework, 924
 - flipped classroom model, 226
 - online systems, 838, 839, 844
 - on vowel dyslexia (VD), 838
- Adaptive competence-based educational system (ACEduSys), 847
- Adaptive enterprise project management (AEPM) capability reference model, 720, 721
- Adaptive learning, 844, 847, 848
- addFirst method, 503
- addLast operation, 501
- Adult learning styles, 230–231
- Advanced Encryption Standard (AES), 519
- AEDP-17 (STANAG 4559), 548, 553
- Agent
 - ABM, 267
 - activities and prices, 967
 - auction, 965
 - crowdsourcing, 972
 - tourist, 962–964, 966
 - tourist activity lists (TA), 965
- Agent-based modelling (ABM), 267, 275
- Agent-based modelling simulation (ABMS), 267
- Agile development
 - ABC, 720, 721
 - active communication and collaboration, 723
 - DevOps (*see* DevOps (development and operations) adoption)
 - operations team, 719
- Agile ITSM
 - adaptation, 924
 - agile software engineering, 924
 - analysis, proffered ITSM frameworks, 931
 - basic agile ITSM framework, 925
 - conceptual review, 922
 - discussion of implications, 931–933
 - FitSM, 928–929
 - frameworks and standards, 922
 - ISO/IEC 20000 standard, 929–930
 - ITIL v4, 925–927
 - lightweight practices, 924
 - manufacturing, 924
 - software engineering, 922
 - VeriSM, 927–928
- Agile practices, 742

- Agile software development
 - and MBSE, 744
 - user stories, 743
 - values, 743
- Alcohol use, 305, 327
- Algebra Project curriculum
 - concept of functions, 19
 - conceptual and procedural knowledge, 18
 - concrete experiences, 24
 - EAGER grant, 18
 - experiential learning, 28
 - experiential mathematics pedagogy, 18
 - foundational events/experiences, 25
 - “hands-on” paradigm, 24
 - linear equations, 19, 20
 - Race Against Time module (*see* Race Against Time module)
 - Road Coloring problem (*see* Road Coloring module)
 - standard representations, 19
- Algorithm
 - analysis, 325–326
 - BST, 321, 326
 - education, 326
 - efficiency, 320
 - radicalization, 775, 776, 778
 - relative location algorithm, 321–325
 - “right-leaning” radicalized recommendation, 775
 - YouTube’s, 775–778
- All India Council for Technical Education (AICTE), 337
- Alternative delivery, 222
- Amazon AWS free IoT services, 211
- Amazon Relational Database Service (Amazon RDS), 579
- American Stock Exchange (AMEX), 731
- Analysis of variances (ANOVAs), 74, 125, 328, 330
- Analytics dashboard
 - desktop application, 279
 - physical therapy (*see* Physical therapy analytics dashboard)
 - pressure sensing fabric technologies, 278
- Anderson’s Interactivity Equation, 118
- Anonymous feedback, 53
- APDS9960 sensors, 204
- Apparent processing time (T_A), 475
- Application programming interfaces (APIs), 72, 197, 279, 283, 284, 551, 659, 665, 796, 797, 854, 901–903
- Application-Specific Integrated Circuits (ASICs), 174
- Arabic language, 838–840, 844
- Architectural aspects
 - HW level virtualization, 452–453
 - reduced power consumption, 453
 - redundancy, 453
- Arduino, 205
- ArrayDeque class, 507
- Array list
 - asymptotic notation, 496
 - Big-Theta, 496, 498
 - class Object, 495
 - generic type invocation, 495, 496
 - instance variable, 495
 - Java program code, 495
 - limitations (*see* Limitations of array list)
 - performance, 496
 - polymorphism, 495
 - software execution time, 463
- Artificial intelligence (AI)
 - algorithms, 290
 - ATM machine, 291
 - autonomous reasoning, 290
 - benefits, 299
 - ethical decision-making processes, 291
 - ethical use, 289
 - fourth industrial revolution, 287, 299
 - negative impact, 288
 - role, 288
 - technological development, 287
 - UN SDG, 291
- Artificial moral agent (AMA), 294
- ASL interpreters, 112
- Assignments/labs/quizzes, 236
- Association for Computing Machinery (ACM), 6
- Asymmetric key encryption, 517
- Asymptotic analysis, 496
- Asymptotic notation, 496, 498
- Asynchronous online delivery, 44
- Asynchronous operation, 480
- Atmospheric water generator (AWG), 730
- Attacking communication wall
 - fully asynchronous operation, 455
 - hierarchic (local) communication, 455
 - internal latency, 455
- Attacking memory wall
 - interrupt and systems calls, context switching, 454
 - register-to-register transfer, 453–454
 - resource sharing without scheduling, 454
 - subroutine call without stack, 454
- Attrition rates, 4, 5, 10, 11, 14
- Auction market, 730, 964–968
 - AMEX, 731
 - electronic, 731

- goals, 731
 - retail/wholesale, 731
 - single auction, 731
 - tourism (*see* Tourism market)
 - Australian Computer Society (ACS), 4
 - Australian software-intensive organization (ABC), 720–723, 725–727
 - Auto-graders
 - computer science course
 - networking, 833
 - operating system, 833
 - cybersecurity science course
 - information security, 834
 - penetration course, 834
 - secure programming, 834
 - database, 827–830
 - empirical data, 835
 - helpdesk course, 833
 - IT course, 833
 - networking admin course, 833
 - online systems, 836
 - programming (*see* Programming auto-graders)
 - Automata theory-based methods, 459
 - Automated guided vehicles (AGV), 535
 - Automatic verification framework, 469
 - Automation, 295, 297, 303
 - Automotive industry
 - AUTOSAR, 557, 559
 - big data, impact, 868–869
 - embedded real-time systems, 557
 - embedded systems modeling languages, 559
 - ERP system, 867, 868
 - industrial revolution, 867
 - innovations, 874
 - AUTOSAR domain-specific language (DSL), 559
 - AUTOSAR Run Time Environment (RTE), 557
 - AUTOSAR standard
 - application layer, 557
 - cooperative component-based development, 557
 - display controller, 560–562
 - integration, 557
 - modeling automotive embedded systems, 559–560
 - overall scheduling, 558, 559
 - partial scheduling, 558
 - PortChain*, 563, 565
 - RTE, 557
 - software layer, 557
 - subsystem development, 557
 - system development, 557
 - temporal requirements, 558
 - AUTOSAR TIMEX, 562
 - AUTOSAR Timing Extensions (TIMEX) model, 558
 - Auto-tuning, 610
 - AWS CodeDeploy, 579
- B**
- Bachelor of Computing (BComp), 5, 10
 - Bachelor of Information and Communication Technology (BICT)
 - ACM, 6
 - ICT courses, 10
 - ICT-related subject, 12
 - second introductory programming unit, 11
 - UTAS, 13
 - Bachelor of Information Systems (BIS), 5
 - Basic Diffie-Hellman, 515
 - Basic Java Programming (BJP), 849, 856
 - Basic Python Programming (BPP), 849, 857
 - Behavior protocol
 - composition and verification algorithm, 464
 - correctness, 460
 - formalism, 460
 - regular expression, 460
 - tokens, 460
 - Belief Rule Base (BRB)
 - data management layer, 242
 - evidential reasoning (ER), 239
 - framework, 238, 241
 - IF-THEN rule, 239, 242
 - input data, 240
 - learning parameters, 238
 - uncertainty, 239
 - Belief Rule-Based Expert System (BRBES)
 - academic knowledge, 240
 - antecedent attributes
 - referential values, 241
 - utility values, 241
 - architecture
 - application layer, 242
 - data management layer, 241
 - interface layer, 242–243
 - AUC data, 243, 244
 - belief degree update, 240
 - belief structure, 239
 - components, 238
 - consequent attributes
 - referential values, 242
 - utility values, 242
 - evidential reasoning (ER), 239

- Belief Rule-Based Expert System (BRBES)
 - (*cont.*)
 - expert assessment level, skill, 243
 - factors, 238
 - framework, 238, 243
 - inference procedures, 239
 - input data, 239
 - input transformation, 239
 - IT graduates' skills proficiency levels, 244
 - ROC, 243, 244
 - rule activation weight calculation, 240
 - rule aggregation, 240
 - skill-level assessment, 243, 244
 - traditional IF-THEN rules, 238, 239
- Berkley FLP program, 219
- Best practices, 39, 40, 72, 131, 275, 874, 927
- Bias, 221, 224, 265, 275, 304, 307, 529
- Bidirectional with On-the-Fly Certification, 518
- Bidirectional with Pre-certification, 518
- Big data, 345, 909
 - with automotive industry, 868–869
 - and ERP (*see* Enterprise resource planning (ERP))
 - essential technology, 863
 - machine learning, 49
 - traditional database technologies, 867
- “Big Data + HPC + Atmospheric Sciences”
 - course
 - HPC Facility, 45
 - innovative approaches, 44
 - 15-module course (*see* 15-Module multidisciplinary course)
 - UMBC students, 45
- BigDecimal class, 506
- Big Science, 49
- Big-Theta notation, 496
- Binary Indexed Trees, 320–321
- Binary search tree (BST), 319–323
- Binary tree, 319, 321
- Bio-inspired models, 487
- Biological systems, 473
- Biology-mimicking architecture, 488
- Biquadratic curve, QRT map, 395, 397, 399
- 1-Bit adder
 - atomic unit, 476
 - circuits, 478
 - classic computing, 477, 478
 - gates, 477
 - idle waiting, 478
 - implementation, 476
 - temporal diagram, 477
 - timing diagram, 477
 - total execution time, 478
 - AND and XOR operations, 477
- Bitcoin, 165, 169, 174, 641
- Blended learning, 121, 124, 951
- BLE Service and BLE Characteristic, 210
- Bletzer's method, 296
- Blockchain
 - Bitcoin, 169, 174
 - “.conf” file, 171, 172, 175
 - constituencies, 164
 - as cryptocurrency, 165 (*see also* Cryptocurrency)
 - file “chainparams.cpp,” 165
 - “genesis block”, 166–168
 - Microsoft Azure portal, 170
 - mining pools, 175
 - networking protocols, 166
 - nodes and ports, 171
 - PoW algorithms, 174
 - P2P network, 169
 - technology
 - application, 641
 - consensus models, 643
 - data management system, 641
 - health domain software, 642–644
 - medicine supply chains, 642
 - National Institute of Standards and Technology, 642
 - smart contract, 643
 - software applications, 641
 - systematic literature review (SLR), 642
 - types of records, 642
- “Block withholding attack”, 175
- Bloom's cognitive and metacognitive learning, 138
- Bloom's revised taxonomy, 821
- Bloom's Taxonomy, 138
- Bloomz*, 355–356
- Bluetooth Low Energy, 210
- BME680 sensors, 204
- BME temperature sensor, 211
- Boiler pressure control real-time system, 465
- Bootstrap, 281
- Bounded floating point (BFP)
 - 80-bit bounded floating point, 366
 - error, 369
 - “exactness” of a value, 370
 - external data sources, 368
 - format, bound field, 367
 - implementation, 367
 - interval, 367
 - under program control, 367
 - range information, 368
 - sizes of formats, 366
 - truncated floating-point value, 368

- zero detection, 366
- BRBES architecture, 241
- Bring your own device (BYOD), 260
- Broom attachment, 111
- Bruner's modes of representations, 24
- Business continuity, 949, 952
- C**
- C++ (programming language), 196, 198, 205, 507
- California State University (CSU), 222
- California State University, Bakersfield (CSUB), 198, 217, 222, 223
- Cambridge Analytica scandal, 289
- Capstone course, 88, 214
- Capstone experience
 - concept report, 60
 - design report, 60, 61
 - integrated curriculum, 61
 - self-and peer-assessment tools, 60
 - survey-type assessment, 60
 - team-based 26-week software development, 59
- T*-test results, 60
 - unpaired *t*-tests, 60
 - Welch's *t*-tests, 60
- Career counseling, 237–238
- Careers-focused strategy, 12
- Case research method, 529
- Case study integrated curriculum
 - formative and summative feedback, 58
 - ICT graduates' technical skills, 57
 - methodology's effectiveness, 57
 - non-technical ICT skills, 57
 - outcomes, 57
 - skill development, 57
- Caucasian, 105
- CC3100 WiFi BoosterPack, 109
- CDT course, 34, 35
- Cellular computing, 449
- Center for Cyber Safety and Education, 31
- Central authority (CA), 518
- Central Processing Unit (CPU), 443
- CGT 270 Data Visualization course, 133, 140
- Change requests (CRs) prediction
 - in aerospace legacy system, 671
 - comparing predictive ability, 683–690
 - curve-fit approaches, 675–677, 681–683
 - vs. defect prediction, 674–675
 - multi-stage approach, 677–678
 - multi-stage approach with TT, 678–680
 - predictions, 671–672
 - SRGM with change-points, 673–674
 - validity threats, 691
- Chess
 - computerized game, 291
 - deterministic two-player primitive recursive game, 421–433
 - as a Gödel number, 424–432
- China mystic supercomputers, 472
- Chi-squared test, 10
- Circular curricular process, 23
- CISCO academy, 81
- CIS RAM (risk assessment process), 513
- Class discussion segments (CDS), 294
- ClassDojo*, 355, 356
- Classical/quantum computing (CQD), 411, 417
- Classic computing, 473
- Class Object, 495, 496
- Classroom environment, 180
- ClassTag*, 356
- Class time, course scheduling
 - cognitive performance, 327
 - methodology, 328
 - sleep loss, 327
 - student academic performance, 327
 - student grades
 - data, 328
 - dependent variable, 328, 330, 333
 - independent samples *t*-test, 328–335
 - instructors, 333
 - morning sections, 330, 333
 - morning vs. afternoon/evening sections, 330, 333
 - student performance, 328
- Clearinghouse, 963, 964
- Client professionalism, 63
- Closed system, 121
- Cloud based software
 - TPS, 769–770, 772
- “CloudCoin”, 165, 175, 176
- Cloud computing, 579–580, 909
- Cloud service-specific security measures, 519
- Cluster addressing, 451
- Cluster head, 448–451
- Cluster members, 449
- Coalition Shared Data (CSD), 547
- CoalitionWarrior Interoperability Exercise (CWIX), 546
- Code execution
 - compatibility with conventional computing, 446
 - cooperation synchronization, 446
 - process, 445–446
 - QT, 444–445
- Cognitive walkthrough (CW) method, 801, 855, 857

- Collaboration competency, 63
- Collaboration skills, 63–64
- Collaboration tools
 - Google Hangouts, 949–953, 957, 958
 - GoToMeeting, 949, 950, 954, 957, 958
 - Microsoft, 954–955
 - primary virtual demands, 949
 - Skype, 949–955, 957, 958
 - Telework, 949
- Commencing domestic ICT students, 13
- Commencing ICT international students, 12
- Common Object Request Broker Architecture (CORBA), 548, 551, 552
- Communication, 352, 353, 487, 488
- Communicational collapse, 485
- Communication mechanisms, 52
- Communication medium, 121
- Communication protocols, 199
- Communication skills, 62–63
- Communications Sector Plan, 70
- Companies
 - crowdsourcing (*see* Crowdsourcing)
 - TPS (*see* Third party software (TPS))
- Competence, 56, 58
- Competence based education, 847–849
- Competence Manager (CM), 795, 797
- Competence Profile (CP), 795
- Compiler, 451–452
- Component-based IoT systems, 465
- Component based software engineering, 560
- Component behavior specification, 460
- Composite operations, 463
- Composition technique, 559
- Comprehensive evaluation, 14
- Computability theory (chess game), 421–424
- Computation
 - ex-machine, 375 (*see also* Ex-machines)
 - turing computable probability, 376
- “Computational and Data Science for All”
 - educational ecosystem, 53
- Computational thinking (CT), 217, 218
- Computer Architecture II, 220, 224
- Computer Engineering program, 103
- Computer Graphics Technology, 133
- Computer processor, 435
- Computer programming, 80
- Computer Science (CS), 6, 55, 825
 - CS Curriculum 2013 report, 71
 - undergraduate program, 32, 34, 41
- Computer Security Incident Response Team (CSIRT), 953
- Computing bottlenecks identification
 - asynchronous operation, 480
 - communication, 487–488
 - high speed serial bus (*see* High speed serial bus)
 - parallelized sequential processing, 485–487
 - synchronous computing, 480
- Computing chain effect/technology/material core, 479
 - digital processing, 480
 - on-chip cache memory, 479
 - position axes, 478
- T_A , 478
 - topologies, 478
- Computing Curricula 2005 Task Force, 72
- Computing curriculum
 - curricular concepts, 72
 - curriculum descriptions, 71
 - gap descriptions, 71
 - HCI, 80–81
 - National University, 69
 - NUS, 70, 73
 - real-world needs, 70
 - recommendations, 81, 82
 - relevance, 72, 73, 75–80
 - revisions, 69
 - survey, 73
 - systems utilized, institutions, 73
- Computing education, 72
- Computing needs, 455
- Computing paradigm, 426, 437, 471
- Computing performance, 487
- Computing professionals, 72
- Computing skills and knowledge, 70
- Computing theory, 472
- Confidentiality, integrity and availability (CIA), 513
- CON group, 223, 227
- Consortium, 962–966
- Content and Knowledge Management System, 795
- Content-based publish/subscribe system
 - event model, 912
 - information distribution mode, 910–911
 - information distribution system, 910–911
 - server, 910
 - wireless dynamic network, 911
- Context switching, 454
- Contingency plan
 - benefits, 938
 - data analysis method, 940
 - data collection method, 940
 - description, 937
 - disaster recovery, 938, 941
 - emergency generators for power, 937
 - escape routes for employees, 937
 - on foreseeable risk, 938

- investigation, 940–941
- methods, 938
- planning for future events, 941
- reliable, 938
- rerouting data, 937
- research approach, 939
- research design, 939
- resource prioritization, 937
- risk assessments, 938
- risks analysis, 937, 938
- RUP, 938
- sampling method, 939–940
- supervisory duties, 937
- technology planning, 939
- Continuous integration and continuous delivery (CI/CD), 546
- Control course, teaching method
 - discrete control systems, 250
 - input/output models, 250
 - lectures, 250
 - MATLAB/SIMULINK computer exercises, 250, 258–259
 - OCD, student case studies, 259
 - real systems, 250
 - software engineering students, 249
 - SYSBOOK platform, 251–255
 - visual interactive demonstrations, 251
 - YOULA parameterization, 250, 255–258
- Control disciplines, 253
- Control education, 260
- Controlling, 10, 62
- Control/non-flipped population, 219
- Control (CON) population, 219
- Convenience sampling, 119, 125
- Conventional computing, 439, 445
- Conventional database framework, 341
- Conventional mathematics representations, 24
- Cooperative development, 558
- Core-to-core register messages, 445
- Corresponding members, 449
- Cost engineering, 880–884, 887, 891
- Cost reduction (CR), 921, 973, 975, 979–983
- Course Authoring Tool (CAT), 793, 797, 798, 805, 806, 850, 851, 855, 857
- Course modules, 32
- Courses learning outcomes (CLOs), 89–90
- COVID-19 pandemic, 44, 115, 949
 - cyber threats, 953–954
 - Google Cloud, 952–953
 - limitations, 958
- Creativity skills, 64
- Critical Structures (Storyboarding Concepts), 124
- Critical thinking skills, 64
- Cronbach’s Alpha analysis, 124
- Cronbach’s reliability analysis, 124
- Crowdsourcing
 - business management method, 971
 - for business strategy
 - masses, 973
 - packager, 972–973
 - professional, 972
 - taxonomy, 972, 973
 - tinkerer, 973
 - costs reduction (CR), 975
 - definition, 971–972
 - improvement of productivity (IP), 974–975
 - innovative activities in companies, 972
 - in Japan, 971
 - procurement of MR, 974
 - purpose, 972, 975–976
 - reasons and effects
 - by business type, 981–982
 - by companies, 979–980
 - by company size, 980–981
 - responding to fluctuations (RF), 975–976
 - SMEs’ management support, 971
 - survey, 972, 976–979
 - via Internet, 971
- Cryptocurrency
 - blockchain technology, 163, 166
 - “CloudCoin”, 165, 175, 176
 - “.conf” file, 171, 172, 175
 - creation and deployment, 164
 - “decentralized security frameworks”, 168
 - elliptic curves, 395
 - Litecoin, 165, 169, 172
 - open-source peer-to-peer project, 164
 - PoW algorithms, 174
 - P2P network, 169
 - scalability and security, 173, 174
 - stages, 164
 - working codebase, 164
- CS careers, 179
- CSCI 549: Intelligent Systems, 185
- CS curriculum, 32, 41
- CS demand, 179
- CSD-Server, 548
- CS education, 179
- CS PhD programs, 186
- CS program
 - central repository, 33
 - courses, 33
 - Cyber Attacks and Defense course, 33
 - cybersecurity program, 32
 - modules, 33
 - Network Defense, 33
 - Network Forensics, 33

- CS program (*cont.*)
 - prerequisites, 33
- CS research identity, 188, 189
- CS undergraduate program, 32
- CS workforce, 179
- CT and FC models
 - alternative delivery, 222
 - CON population, 219
 - data collection
 - not reported, 221
 - reported, 221
 - FC population, 219–221
 - implementation, 222
 - instructors considerations, 222–223
 - qualitative preliminary results
 - CON group, 223
 - FC group, 224
 - quantitative comparison, 224–227
- CT framework, 218
- Curling Canada, 114
- Curricular concepts, 72
- Custodian Support Teams (CSTs), 545
- Customer relationship management (CRM)
 - easy-to-use features, 706
 - technical debt (*see* Technical debt, CRM application)
- Cyber Attack and Defense (CAD), 33
- Cyber-attacks, 288, 290
- Cyber-aware professionals, 31
- Cyber defense, 31, 290
- Cyber Defense Track, 42
- Cyber ethics, AI
 - algorithms, 290
 - ATM machine, 291
 - automation, 295
 - autonomous reasoning, 290
 - business, 291
 - Cambridge Analytica, 289
 - citizenship of machines, 299
 - CSD, 294
 - cyber-attacks, 288
 - cyber security, 288
 - ethical decision-making processes, 291
 - Facebook, 289
 - Google, 288
 - iGens, 289–290, 298, 299
 - IT, 291
 - journal-posts students, 300
 - pro-artificial intelligence, 300
 - pro-autonomous devices, 300
 - response distribution, 292
 - response rate, student decisions, 297
 - sample, journal posts, 294–297
 - sample size
 - lecture, AI, 292
 - SRJ, 292–293
 - student choice, CDS, 296
 - students, 292
 - teaching, 293
 - UN SDG, 291
 - Word Clouds, 30
 - words/strings of words, online journals, 297, 298
- Cyber Fellows, 825
- Cyberinfrastructure (CI), 44
- Cyber-physical systems (CPS)
 - CPSoS, 511
 - DEIS project, 512, 521
 - dependability, 511
- Cyber-physical systems of systems (CPSoS), 511
- Cybersecurity, 288
 - autograders use, 834–835
 - computer science, 825
 - courses, auto-graders
 - information security, 834
 - penetration course, 834
 - secure programming, 834
 - for TPS, 770–772
 - Zoom, 956–957
- Cybersecurity-aware CS graduates, 32
- Cybersecurity awareness, 35, 39
- Cybersecurity career interest
 - cyber attacks and defense, 39–40
 - network defense, 39–40
 - network forensics, 41
- Cybersecurity core curriculum, 32
- Cybersecurity education programs, 32
- Cybersecurity hiring crisis, 41
- Cybersecurity knowledge
 - cyber attacks and defense, 37
 - network defense, 37
 - network forensics, 37
 - Operating Systems, 37
 - SQL injection, 37
- Cybersecurity practices awareness
 - countermeasures against attack, 39
 - cyber attacks and defense, 38
 - network defense, 38–39
 - network forensics, 39
- Cybersecurity preparedness, 32
- Cybersecurity-related courses, 41
- Cybersecurity-related key concepts, 38
- Cyberspace, 290
- Cyber threats, 31

D

- Data access and reading, 498–500
- Data and information literacy
 - challenges, 131
 - data visualization course, 131
 - Dear Data (*see* Dear Data project)
 - problem-based learning, 132
 - students, 131
 - teaching, 132
 - visually communication, 138
- “Data + Computing + X” course, 43
- Data confidentiality, integrity and availability (CIA) triad, 513
- Data-driven science, 43
- Data-intensive systems
 - challenges, 577
 - cloud computing, 579–580
 - cluster, 592
 - deployment environment, 584
 - dynamic scaling methodology (*see* Dynamic scaling methodology)
 - earth observation datasets, 578
 - EASTWeb application, 578
 - EASTWeb system, 584, 592
 - fields, 577
 - horizontal scaling capability, 592
 - implications, 577
 - MapReduce, 592
 - methodology
 - database transformation, 583
 - deployment environment, 582–583
 - modifying, 581–582
 - scaling, 580–581
 - systems processing, 577
 - virtualization, 592
 - web application, 591
- Data literacy, 132
- Data management, 347, 867, 869, 872, 873
- Data Mine*, 133
- Data Mine Data Visualization Learning Community*, 133
- Data Mine Learning Community Initiative*, 133
- Data mining
 - accuracy, 308
 - confusion matrix, decision tree, 310
 - dataset size reduction, 315
 - factors, 317
 - IBk Nearest Neighbor, 307–309
 - J48 Decision Tree, 307–309, 313
 - Logistic regression, 316
 - Multilayer Perceptron, 307, 308, 313
 - predictivemodels, 303
 - Random Forest, 307, 308, 311, 317
 - Random Tree, 307, 311–313
 - student GPA dataset, 303, 306
 - ZeroR (baseline) classifier, 308
- Data security, 340, 347, 348, 513, 771
- Data structures
 - array list (*see* Array list)
 - doubly linked list, 500–501
 - implementation, 494
 - linked list, 498–500
 - memory utilization, 494
 - organized collection, 495
 - performance evaluation, 494
 - performance gap, 493
 - queue, 506–507
 - stack, 505–506
- Data visualization, 341
 - capacity building, 138
 - pedagogy, 138
- DDI applications, 519
- DDI packages, 515
- DDI security protocol at rest
 - AES, 519
 - application security, 520
 - asymmetric encryptions, 519
 - cloud service-specific security measures, 519
 - cloud “Storage Service”, 519
 - database security, 520
 - encryption key storages, 519
 - HSMs, 519
 - local memory, 518
 - symmetric encryptions, 519
 - symmetric keys, 519
- DDI security protocol in transit
 - platoon use case, 516–518
 - between system components, 515
 - system to cloud server, 515
 - system to system, 516
- Deaf curlers, 104
- Deaf curling athletes, technology solutions
 - assignments, 108
 - conceptual frameworks, 104–107
 - course design, 108–109
 - design project, 104
 - nontechnical material, 113
 - perceptions, 113
 - students’ designs, 110–113
 - sweeping signals, 104
 - wayfinding, 114
- “Dear Data Artifact”, 136
- Dear Data Assignment, 133–138
- Dear Data postcard visualization assignment
 - methodology
 - administrative tasks, 135
 - artifact, 136

- Dear Data postcard visualization assignment
 - methodology (*cont.*)
 - assessment (*see* Postcard assignment assessment)
 - CGT 270 Data Visualization course, 134
 - class assignment, 134
 - data literacy, 135
 - electronic devices, 134
 - information literacy, 135
 - instructional perspective, 138
 - introductions theme, 135
 - limitations, 139
 - postcard template, 134
 - visual communication, 133
 - visual human portrait, 135
- Dear Data project
 - data visualization course, 132
 - educational and social analysis, 132
 - elements, 132
 - participants, 133
 - postcard visualization assignment (*see* Dear Data postcard visualization assignment methodology)
 - reinforce content, 133
 - student self-assessment, 132
- Debugging skills, 201
- Decision-making assignments, 120
- Decision Trees, 307–309, 313, 316
- Deep learning, 46, 472
- DEIS project, 512, 521
- Deletion operations performance
 - ArrayList, 505
 - execution time *vs.* data size, 504
 - LinkedList class, 505
 - maximum heap size, 505
 - memory utilization *vs.* data size, 504
 - performance gap, 504
 - removeFirst method, 505
 - removeLast method, 503, 504
- Demonstrations, 235
- Denial of service (DOS), 513
- 'Denied' signal, 452
- Dentacoin*, 641
- Department of Homeland Security (DHS), 32
- Department of Social Justice, 344, 347
- Department of Social Justice and Empowerment of Minorities, 347
- Desalination plants, 731
- Design cycle, 105
- Desktop publishing tools, 78
- Deterministic two-player game, 421
- Developer framework, 206
- Development Strategic Plan, 53
- DevOps (development and operations)
 - adoption
 - case study in ABC (*see* DevOps case study in ABC)
 - definition, DevOps, 720
 - factors, 719–720
 - integration, 720
 - interesting approach, 719
 - and Microservices Architecture, 726
 - real-time, 720
 - security, 727
- DevOps case study in ABC
 - AEPM capability reference model, 720–721
 - iteration management
 - gaming platform, 720, 722
 - iteration implementation, 723–725
 - iteration team, 722
 - post-iteration (heuristics), 725–726
 - pre-iteration, 722
 - release cycle, 721
 - microservices, 727
- Didactical Structural Template Manager (DSTM)
 - Competence Manager, 797
 - CW method, 801
 - functionality, 800, 802
 - functional requirements, 800
 - implementation, 800, 807
 - integration, 800
 - Learning Designer, 797
 - Teacher, 797
 - use case diagram, 798
- Didactical structural templates (DST)
 - existing DSTs, 802
 - export, 796
 - gamified moodle course, 806–807
 - implementation, DSTM, 800, 802
 - on IMS-LD, 794
 - local and security instruction, 802
 - LP and pedagogical structure, 794
 - Moodle course, creation, 805–806
 - new DST
 - complete defined DST, 803, 804
 - with defined play, 803, 804
 - initial content, 803
 - as ST, 794
 - tool/ production environment, 795
- Digital Dependability Identity (DDI)
 - assurance cases, 512
 - CIA, 513
 - components/system, 512
 - dependability data models, 512
 - interrelation, 512

- as modular assurance cases, 512
 - ODE, 513
 - risk assessment process, 513
 - security assurance, 512
 - for security assurance, 512
 - security protocol (*see* DDI security protocol
 - at rest; DDI security protocol in transit)
 - subcomponents, 512
 - system's model-based safety reflection, 512
 - targets, 512
 - Digital goods, 880, 881
 - Digital natives, 289, 290
 - Digital Systems Design, 103
 - Digital transformation
 - collaboration (*see* Collaboration tools)
 - ITIL v4, 925
 - in procurement, 879–880
 - VeriSM, 927, 933
 - Directorates of technical educations (DTEs), 337, 347
 - Display controller, AUTOSAR
 - informal requirements, 560–562
 - overall scheduling, 562
 - partial schedulings, 562
 - subsystems, 562
 - system structure, 560–562
 - Distributed agile, 720, 723, 725, 726
 - Distributed ledger technologies (DLTs), 570
 - Distributed systems, 742
 - Distribution system operator (DSOs), 569
 - Docker, 285, 827, 829, 833
 - Document-based approach, 742
 - Domestic course attrition, 11
 - Doubly linked list
 - data access and reading, 500
 - deletion operations, 501
 - insertion operations, 501
 - java class, 500
 - nodes, 500
 - Drug use, 304, 316
 - Duration automata, 459, 461
 - Dynamic personalized learning path (DPLP), 856, 857
 - Dynamic scaling methodology
 - database transformation, 589–590
 - helper project algorithm, 584–587
 - IMERG_Project, 590, 591
 - system modification
 - base version, 587–589
 - virtual machine (VM) version, 589
 - Dynamic variable, 439, 445
 - Dynamic wireless network, 910
 - Dyslexia
 - adaptive online education systems in
 - Arabic, 839–843
 - coronavirus pandemic, 840
 - definition, 837
 - diagnosis, 840
 - research, 844
 - TrainDys system, 838, 840, 841, 843
 - types, 838
 - VD and SVD, 838
 - Dyslexics, 837–841, 843
- E**
- EAGER grant, 18, 28
 - Early-Concept Grant for Exploratory Research (EAGER), 17
 - Earth-atmosphere radiative energy balance, 46
 - EAST-ADL2 modeling language, 559
 - ECE 3760 Digital Systems Design, 103
 - e-Commerce, 899
 - e-Competence Framework (e-CF), 849, 850
 - Ecosystem, 230
 - Edge computing, 909
 - Education, 290, 345
 - Educational data mining, 303
 - Educational institutions, 345
 - Educational process, 72
 - Educational services, 781, 782, 784, 788, 790
 - Educational system, 229
 - E-learning
 - auto-graded assignments (*see* Auto-graders)
 - environment, 118
 - GOAL, 826, 827
 - Gradescope, 827
 - SIMnet, 827
 - Electronic auction market, 731
 - Electronic commerce, 962
 - Electronic market, 961, 962, 964
 - ElectronJS, 281
 - Elevator pitch, 199
 - Elliptic-Curve Diffie Hellman (ECDH), 515
 - Elliptic curve method (ECM)
 - Edwards curve, 396
 - factorization with QRT maps
 - biquadratic curve, 398, 401, 402
 - Lyness map, 401, 403–405
 - Somos-4 QRT Map, 401–402
 - Somos-5 QRT Map, 403
 - symmetric QRT map, 399, 400
 - implementations, 397
 - Pollard's rho method, 395
 - scalar multiplication, 405–406
 - Weierstrass cubic, 396
 - Elliptic curves, 395–397, 401, 406

- Elliptic divisibility sequence (EDS), 397, 401
- Email communication, 112
- E-marketplaces, 962
- EMPA-aware code, 446
- EMPA-based computing, 442
- EMPA Communicating Element (ECE), 443, 444
- EMPA communication, 450
- EMPA core
 - EME, 444
 - EPE, 443
 - ESME, 444
- EMPA implementation
 - 'ad hoc' structures, 446–448
 - code execution, 444–446
 - communication, 450–451
 - compiler, 451–452
 - conventional computing, 443
 - conventional many-core processors, 442
 - the core, 443–444
 - cores clustering, 448–450
 - free resource, 442
 - processor, 448
 - true parallelism, 442
 - variable granularity, 443
- EMPA Morphing Element (EME), 443, 444
- EMPA new features
 - architectural aspects, 452–453
 - attacking communication wall, 454–455
 - attacking memory wall, 453–454
- EMPA Processing Element (EPE), 443, 444
- EMPA processor, 452
- EMPA Storage Manager Element (ESME), 443
- Empathize*, 107, 108
- Empirical analysis, 6
- Employability skill set, 56
- Employment-oriented curriculum design, skill
 - development training program
 - adult learning styles, 230–231
 - constraint identification, 231
 - evaluating student performance, 232
 - learning goals and outcomes, 231
 - students' educational/functional skills deficits, 230
 - utilitarianism, 231–232
- Empowerment of Minorities, 347
- Enactive-iconic representation, 24
- "Encryption at Rest" service, 520
- Encryption key, 516
 - storages, 519
- 'End of code fragment' code, 445
- End-to-end encryption, 951, 952, 956, 957
- Energy-efficient intelligent vehicles, 873
- Energy flexibility marketplace
 - analysis
 - clients' market engagement, 574
 - market operating, 571
 - Market Operation category, 574
 - Market Participation category, 573
 - number of appearances, 571, 572
 - requirements, 573
 - Settlement subtopic requirement, 574
 - Settlement topic group, 571
 - validation, 574
 - business decision, 575
 - business-related design, 576
 - client participation, 575
 - design phase, 568
 - digital currencies, 576
 - electricity, 567
 - equipment control, 568
 - expert panel, 570–571
 - FLEXIMAR
 - architecture, 569–570
 - platform, 568
 - machine-to-machine economy, 576
 - market institution, 567
 - power grids, 567
 - requirements, 567, 571
 - service design, 575
 - software system, 567
 - terms of service (TS), 571
 - worksheet design, 571, 572, 575
- Energy-wasting solutions, 435
- Engineering design cycle, 105
- Engineering design process, 106
- Engineering education, 249
- English language learners (ELLs), 360
- Enhanced operators, 462
- Enhanced time behavior protocol (ETBP)
 - applications, 465–469
 - classical regular expressions, 461
 - composition, 463–464
 - duration automata, 461
 - enhanced operators, 461, 462
 - !interface.method, 461
 - IoT systems, 461
 - programs model systems' interaction
 - behavior, 461
 - time consumption constraint, 461
- Enterprise-level software, 705, 709
- Enterprise resource planning (ERP), 705
 - automotive companies, 863
 - with big data, 867–868, 872
 - complexity, ERP responsiveness, 871, 874
 - data management, 869
 - influence on automotive industry, 868
 - information systems, 865

- SIST model, 873
 - SLR, 864–868 (*see also* Systematic literature review (SLR))
 - trust issues, 867, 869, 871, 873–874
 - Environmental sustainability, 105
 - Epidemiological Applications of Spatial Technologies (EASTWeb) system, 577
 - Equity, 107
 - Equity, diversity and inclusion (EDI)
 - modelling and simulation (*see* Modelling and simulation, EDI)
 - quantitative approach, 266
 - STEM fields, 265
 - Erasable programmable read-only memory (EPROM), 519
 - Ergonomics, 111
 - Error range, 686
 - Error/small programming, 213
 - ESP32 program, 211
 - ESP8266 Wi-Fi chip, 202
 - Essential instructional elements, PeopleNTech
 - demonstrations, 235
 - hands-on class labs, 235
 - student public speaking via classroom presentation project, 235
 - traditional lecture method with audio-visual aids, 234
 - tutoring, 235
 - ETBPSV tool, 464, 466, 467
 - Ethereum, 641
 - Ethical theories, 293
 - EU planned supercomputers, 472
 - European Qualification Framework (EQF), 849
 - Evidential Reasoning (ER), 238, 239
 - Exactness, 366–372
 - Exact subtraction, 370–372
 - Exclusion criteria, 513
 - Ex-machines
 - computing languages
 - ex-machine $Z(x)$, 382–388
 - turing incomputable properties, $Z(x)$, 386–388
 - information-theoretic analysis, 389
 - meta and random instructions, 375, 379–381
 - non-autonomous dynamical system, 375
 - random instructions, 377–379
 - standard instructions, 377–379
 - Turing machine, 375
 - Turing’s halting problem, 388–389
 - Explicitly many-processor approach (EMPA)
 - advantages, 441–442
 - code execution, 440
 - denied cores, 440
 - general principles, 437–439
 - implementation (*see* EMPA implementation)
 - meta-instructions, 440, 441
 - parallel processing, 439
 - PU’s, 439
 - QT , 439
 - two-layer processing, 440
 - Explicit MultiThreading (XMT), 444
 - Exploratory measures, 189
 - Extract Class refactoring
 - agglomerative clustering algorithm, 698
 - application, 702–704
 - circle, 695, 696
 - Circle and GeometricCircle, 697
 - clients, 695, 696
 - definitions, 699
 - GeometryApp*, 696
 - GraphicsApp*, 696
 - ISP, 697
 - Jaccard distance, 698
 - Max-Flow Min-Cut algorithm, 698
 - object-oriented systems, 695
 - proposed approach, 699–702
 - software systems, 695
 - structural and semantic similarities, 698
- F**
- Face-to-face
 - classes, 825
 - course, 50
 - teaching, 50, 51
 - Failure propagation modeling, 512
 - Fault-tolerant systems, 453
 - FC experience, 224
 - FC group, 224, 227
 - FC population
 - class performance, 221
 - complete active learning/CT activities, 220
 - Computer Architecture II, 220
 - CON group’s online homework, 220
 - JiTT, 219
 - online assessment/quiz, 219
 - pair programming, 220
 - traditional classroom, 220
 - FD (university teacher professional training)
 - activities, learnings and worksheets, 815
 - activity of improvement, 809
 - classroom observation, 810–811
 - cognitive and metacognitive questions, 811–813

- FD (university teacher professional training)
(*cont.*)
enhancing cognition and metacognition,
815–817
“Grassroots FD Project 2019”, 809–822
ICE model, 813–814
“mandatory duty”, 809
metacognition, 813
process of reflection, 813
systematic engagement, 809
teachers’ learning, 818–819
- Federal Big Data Research, 53
- Federal Cybersecurity Workforce Strategy, 31
- Financial assistance management
conventional database framework, 341
data visualization, 341
framework description, 343–344
hardware and software specifications, 344
intelligent and immediate feedback, 341
new framework implementation, 341–343
performance prediction, students, 340–341
risk detection, 340
- Financial assistance schemes, 338
- Financial assistance students, Punjab Technical
Education system
AICTE, 337
big data, 345
caste-based population, 338, 339
certificate level skill/vocational courses,
337
community-based population, 338, 339
data security, 348
data security risks, 347
educational institutions, 338
financial assistance management (*see*
Financial assistance management)
import data to MongoDB
checking import data, 345
import CSV file, 344–345
- INR
year-wise OBC students and claims,
347
year-wise SC students and claims, 347
- MapReduce framework, 346
- MongoDB, 337, 346, 348
- NoSQL databases, 337
- other backward class (OBC), 337, 346,
347
- PMS scheme, 346
- scheduled caste (SC), 337, 346,
347
- scholarship schemes, 338, 340
- social justice schemes, 338
- social welfare schemes, 338, 339
- UGC, 337
universities, 338
- First-in-first-out (FIFO), 506
- First-in-last-out (FILO), 505
- Fisher College, 825
- FitSM (proffered agile ITSM frameworks),
922, 924, 928–929, 931–934
- Flask (web application framework), 283, 284,
901, 906
- Flipped classroom (FC), 217, 218
- Food and Drug Administration (FDA), 641
- Forbes Innovation*, 32
- Ford-Fulkerson algorithm, 341–342
- Formal modeling methods, 459
- Formal representations, 23
- Foundational programming pass rates, 13
- Foundational programming skills, 9
- Foundational programming unit, 9
- Fourth industrial revolution (4IR), 229, 230,
287
- Fourth space dimension, 473
- Fragmentation, 497
- “Free” processors, 455
- FreeRTOS, 205
- Functional and physical prototypes, 114, 115
- The *future light cone*, 474
- Fuzzy matching algorithm
in content-based publish/subscribe, 910,
912–916
- G**
- Game-changing technology, 741
- Games and Creative Technology (GCT) major,
7, 12
- Games-career-focused major, 12, 13
- Game theory, 421–433
- Gas price, 652
- GCT major, 12
- General-purpose computing systems, 472
- GeneralStore* approach, 560
- Generation Y, 289
- Generation Z, 289
- Generic Attributes (GATT), 210
- Generic LinkedList class, 501
- Genesis block, 166–168
- GitHub, 159, 164, 167, 357, 723, 724, 903, 946
- Global climate model (GCM), 44, 47
- Gödel numbers, 421–424
- Good communication medium, 121
- Google Hangouts, 949–953, 957, 958
- Google Meet (GM), 952
- Google quantum supremacy (GQS), 412, 414,
416, 417

- Google Quantum Team's methods, 514–517
 GoToMeeting, 949, 950, 954, 957, 958
 GPA grade distribution, undergraduate computer science students
 attribute distribution, illicit drug use, 306
 attributes use, survey, 304–3010
 attribute values, 316
 class distribution, 306
 data mining (*see* Data mining)
 Decision Trees, 316
 drug use, academic performance, 304, 316, 317
 evaluation metrics, 308
 feature extraction, 316
 hours spent, social media platforms, 313, 314
 illicit drug use, 313–315
 name and description, attributes, 305
 personality, 304
 sleep, 303, 317
 sleep deprivation, 304
 sleep quality, 304
 social media use, 304, 306, 313, 317
 study hours per week, 313–315
 Grade Point Average (GPA), 120, 125–127, 303
 See also GPA grade distribution, undergraduate computer science students
 Gradescope, 827, 834
 Gradience Online Accelerated Learning (GOAL), 826, 827
 Grading process, 199
 Graduate competency, 6
 Graduate employment outcomes, 56
 Graduate ICT career outcomes, 6
 Graphics tools, 79
 Graph theory, 319
 “Grassroots FD Project 2019”, 809—822
 Grounding metaphors, 25
- H**
 HAAT model, 107
 Hadoop Distributed File System (HDFS), 47
 Halting problem, 388–389
 Hands-on experience, 40
 Hands-on hardware projects, 198
 Hands-on lab assignments, 198
 Hard skills, 81
 Hardware-based approach
 actuators, 205
 challenges, 201
 computing platform, 201
 microcontrollers, 202–204
 peripherals, 205
 programming framework, 205–206
 sensors, 204–205
 Hardware experience, 200
 Hardware security modules (HSMs), 519
 Hardware skills, 199–200
 Harper Archer Middle School in Atlanta, 26
 Hazard and risk analyses (HARA), 512
 Heap memory, 502
 Helpdesk course auto-graders, 833
 Hierarchic (local) communication, 455
 Higher Education Institutes (HEI), 793
 Higher-education institutions, 31
 Higher-education system, 131
 High performance computing (HPC), 44–48, 53, 720
 High Performance Linpack (HPL) benchmark., 455
 High speed serial bus
 bus bandwidth, 484
 communication, 483, 484
 design principle, 484
 emergency measures, 484
 foreign contribution, 482
 formalism, 482
 neuromorphic systems, 481
 neurons, 482
 operation, 483
 packages, 481
 parallel buses, 481
 technical implementation, 481
 temporal behavior, 482
 temporal logic, 483
 transfer time, 484
 transmission, 484
 ‘Hiring core’, 444, 446
 Hiring policy
 academia, 266
 EDI, 265
 engineering faculty hiring process, 266
 engineering practice, 265
 engineering profession, Canada, 265, 275
 modelling and simulation (*see* Modelling and simulation, EDI)
 Historically black colleges and universities (HBCUs), 48
 Homework and JiTT quizzes, 221
 Honors program
 academic excellence, 277
 civic learning and leadership, 277
 continuous learning, 277
 global citizenship, 277
 integrity, 277

- “How to Read” section, 134, 135, 139
 - HPC facility, 44
 - Human-centric problem statements, 109
 - Human computer interaction (HCI), 80–81
 - Human Computer Interaction course (HCS386), 78
 - Hypertext Transfer Protocol Secure (HTTPS), 515
- I**
- IBk Nearest Neighbor, 307, 309
 - ICE model (vocal learning), 813–816, 818–822
 - ICT career, 4, 6–8
 - ICT curriculum
 - amending misconceptions, 6–8
 - career-based pathways, 13
 - careers-focused strategy, 12
 - improving academic success, 8–9
 - improving engagement, 8
 - improving perceptions and motivation, 6–8
 - knowledge and skills, 55
 - longitudinal study, 6
 - method, 9–10
 - professional skills, 6, 55
 - renewal design process, 6
 - renewal strategy, 71
 - research question, 55
 - strategies, 6
 - student commencements, 10–12
 - ICT domestic students, 11
 - commencement rates, 13
 - ICT graduates, 56
 - technical skills, 6
 - ICT higher education courses, 7
 - ICT international student commencement rates, 13
 - ICT professionalism, 9
 - ICT Professional major, 7
 - ICT-related subject, 7, 9, 12
 - ICT student commencements
 - academic success, 11
 - alternative entry point, 11
 - BComp/BIS course, 10
 - BICT course, 10
 - course attrition rates, 10, 11
 - domestic students, 11–12
 - double counting, 10
 - foundational programming, 11
 - international students, 12
 - IDE and toolchain setup, 207
 - Identity Management for Access Control (IMAC), 519
 - Idle waiting time, 474, 476
 - IEEE Computer Society, 85
 - IEEE floating-point standard, 365–366
 - iGens, 289–290, 298, 299
 - Illicit drug use, 305, 309, 312
 - ‘low power’ state, 453
 - Imaginary cryptocurrency, 165
 - Implicit hardware/software contract, 436, 473
 - Improving academic success, 8–9
 - Improving student engagement, 8
 - IMS Learning Design (IMS-LD), 794–797, 807
 - new XML schema tree of component, 799
 - specification, 797
 - XML schema tree, 799
 - Incomputable, Turing, 376, 382, 386–388, 390
 - Incremental development methods, 455
 - Independent samples t-testing, 74
 - Independent software vendors (ISVs), 706, 709
 - Index identification, 186
 - Indian higher education system, 337
 - Industrial Training Institutes (ITI), 337
 - Inexact subtraction, 370–372
 - Inferential analysis, 74
 - Informal language, 23
 - Information and communication technology (ICT), 55, 70, 289
 - career, 4
 - commencements and attrition, 4
 - courses, 4
 - curriculum, 3, 6
 - graduates, 3, 4
 - higher education, 3
 - student perceptions, 7
 - students, 5
 - undergraduate student population, 4
 - Information dissemination, 971
 - Information integrity, 340
 - Information literacy, 131, 132
 - Information Requirements Management and Collection Management (IRM&CM), 547
 - Information security-specific courses, 32
 - Information systems (IS), 6, 55, 80
 - Information Technology (IT), 55
 - Information Technology Curricula 2017, 71
 - In-house training programs, entry-level IT graduates, 229
 - Initial statechart, 111
 - Innovational approaches, 88
 - Input/Output (I/O) instructions, 436
 - Insertion operations, linked list, 499
 - Insertion operations performance
 - addFirst method, 503
 - addLast operation, 502

- Double data type, 503
- execution time vs. data size, 502
- GC, 502
- integer, 502
- Java Runtime class, 502
- Java System class, 502
- JVM, 503
- maximum heap size, 502
- memory usage, 502
- memory utilization vs. data size, 503
- nodes, 502
- Instance-based classifier, 307
- Institute of Electrical and Electronics Engineers (IEEE)
 - ISO/IEC/IEEE 60559, 365
- Instructor-driven WhatsApp groups, 118
- Integrated curriculum, 57
- Integrating professional skill development,
 - ICT curriculum
 - capstone experience, 58–61
 - collaboration competency assessment, 58
 - collaboration skills, 63
 - communication skills, 62–63
 - competency, 58
 - controlled longitudinal study, 62
 - creativity skills, 64
 - critical thinking skills, 64
 - empirical evaluation, 58
 - employability, 65
 - KIT105 ICT Professional Practices, 58, 59, 61–62
 - students' behaviour, 58
 - teamwork ability, 58
 - technical skills, 64–65
 - t-tests, 59
- Intelligent feedback, 341
- Intelligent Systems, 182
- Interaction Equivalency Theorem, 118, 121, 122
- Interactivity Equivalency Equation, 120
- Interconnection cache, 455
- Inter-Core Communication Block (ICCB), 451
- Interface Segregation Principle (ISP), 697
- Intermediary, 163, 169, 963
- Internal Model Control (IMC), 256
- Internal signal propagation, 448
- International female Master's students, 182
- International Personality Item Pool (IPIP scale), 600
- International student course attrition rates, 14
- Internet, 117
- Internet age, 289
- Internet of Things (IoT), 195, 909
- Internships, 232
- Intervention class, 183
- Intervention course, 187
- Intrinsic personality traits
 - Big Five model, 598
 - data analysis
 - age group, 601, 605
 - average and standard deviation, 601, 603
 - clusters, 601, 607
 - company size, 601, 605, 606
 - correlation coefficients, 601, 603
 - distribution of scores, 601, 602
 - education level, 601, 604
 - p*-values, 601, 603
 - role type, 601, 606
 - sample sizes, sex, 601, 605
 - trait scores, sex, 601, 604
 - data collection, 599–601
 - groups, 598
 - hypothesis, 597
 - MBTI, 598
 - technical personality, 597, 599
 - testing methodology, 599–601
- IoT concepts, 198
- IoT courses
 - advanced undergraduates, 196
 - California State University system, 195
 - challenge, 195
 - hardware and kits, 196
 - students and educators, 195
- IoT devices, 200
- IoT projects, 200
- IoT prototyping projects, 202
- IoT teaching approaches
 - audience and assumptions, 198
 - course structure, 198–199
 - hardware-based (*see* Hardware-based approach)
 - interdisciplinary, 197
 - interdisciplinary field, 196
 - introducing key hardware skills, 199–201
 - labs (*see* Lab experiments, IoT teaching)
 - learning objectives, 197
 - microcontroller, 197
 - REST APIs, 197
- ISO/IEC 11770 key management standard, 514
- ISO/IEC 20000–1:2018 standard, 921, 922, 924, 928–934
- ISO/IEC 27002 standard, 514
- ISO/OSI 7-layer network, 38
- IT curriculum, 6
- ITIL v4 (proffered agile ITSM frameworks), 922, 925–927, 930–934
- IT-related technical unit, 8

- IT service management (ITSM)
 - CMMI-SVC v1.3, 921
 - definition, 923
 - frameworks and standards, 921
 - ITIL v2011, 921
 - IT management domain, 922
 - IT operation processes, 922
 - service management, 923
 - utilization, 921
 - value, 923
- IT training, 231

- J**
- Java class, 498, 500, 666
- Java Collections framework
 - academia and research, 494
 - asymptotic analysis, 493
 - data structures (*see* Data structures)
 - programming languages, 507
 - software applications, 493
 - space complexity, 493
 - stack class, 507
 - time complexity, 493
 - unified architecture, 493
- Java program code, 495
- Java programming, 966
- JavaScript Object Notation (JSON), 796
- J48 Decision Tree, 307, 309, 313
- JISR Test Center, 549–550
- JiTT style quizzes, 222
- Job placement services, 237–238
- Joint Intelligence, Surveillance and Reconnaissance (JISR), 545, 547, 554
- Just-In-Time-Teaching (JiTT), 219

- K**
- Kahoot! quizzes, 221
- Keil IDE, 109
- Key Management Service (KMS), 516
- Key model
 - confidence, 619
 - creation, 613–616
 - note selection, 618
 - pitch correction, 624
- KIT105 ICT Professional Practices, 61, 62
- Knowledge-Management Ecosystem Portal (KM-EP), 793, 800
- Knowledge units (KUs), 33
- Krumhansl/Schmuckler algorithm, 614, 618

- L**
- Lab experiments, IoT teaching
 - additional activities, 212
 - communication part 1, 209–210
 - communication part 2, 210
 - communication part 3, 210
 - IDE and toolchain setup, 207
 - management, 211
 - preparation, 206
 - security, 211
 - sensors, 207–209
 - visualization, 211–212
- Last-in-first-out (LIFO), 505
- Learner satisfaction, 781, 782
- Learner’s opinions in online learning platforms
 - big data, 781
 - from course perspective
 - average rating score of a course, 785
 - course category analysis, 784
 - opinion modality analysis, 783–784
 - opinion-rating relation, 785
 - exploratory analysis, 789
 - from instructor perspective, 787–789
 - large-scale course platforms, 789
 - from learner perspective, 786–787
 - learners’ attitude, 791
 - learning analytics, 782–784, 786–789
 - perspectives, 782
- Learning Design API, 796, 801
- Learning disability, 837
- Learning management system (LMS), 117, 118, 120, 292, 850
 - learning content, HEI, 793
 - via LTI, 807
- Least-squares, 615
- LED bank, 110
- Legacy systems, 671–673, 680, 686, 691
- Legal compliance, 770–773
- Likert item analysis
 - improved social interaction, 226, 227
 - instructor confidence perceptions, 226
 - online/advanced traditional classroom, 225
 - quantitative results, 228
 - student performance, 227
 - subset analysis, 225
 - tendencies, 226
- Likert items, 221, 224, 225
- Limitations of array list
 - deletion at a beginning, 497
 - deletion at a end, 498
 - deletion at a specified index, 497
 - insertion at a beginning, 497

- insertion at a specified index, 497
- reallocation, 496
- time complexity, 496
- LINDDUN (Computational Intelligence), 521
- Linear transformations, 23
- Linked list
 - data access and reading, 498, 499
 - deletion operations, 500
 - implementation, 494
 - insertion operations, 499
 - Java class, 498
 - maximum size, 499
 - memory fragmentation, 498
 - nodes, 498
 - object references, 499
 - performance, 494
- Linux, Apache, MySQL, PhP (LAMP), 209, 212
- Local Area Network (LAN), 77, 441
- Logistic Regression, 316
- Long Range (LoRA), 204
- Long-range interneurons, 488
- Long-short-term memory (LSTM), 47
- LoRa capabilities, 210
- LoRa encryption approaches, 211
- LoRa library, 211
- LoRA vs. LoRaWAN communication, 210
- LTSA tool, 464
- Lyness map, 397, 401, 403–407

- M**
- Machine learning, 909
- Magnitudes, 22
- Management resources (MR), 973, 974, 979–982
- Manual physical therapy, 279
- Map-based publish/subscribe system, 912
- MapReduce framework, 346
- Massive Open Online Courses (MOOCs), 750, 782
- Master’s level CS courses, 189
- Master’s students, 180, 182
- Master’s students’ research identity/self-concept, 192
- Mathematical-style pseudo-code, 342
- Mathematical symbols, 27
- Mathematics-based theory, 471
- MATLAB/SIMULINK computer exercises, 258–260
- MediBloc*, 641
- Medical Device Directives, 641
- Medical device software (MDS), 643

- Melody-based pitch correction model
 - audio pitches, 609
 - auto-tuning, 610
 - challenges, 609
 - goal, 609, 624
 - Krumhansl/Schmuckler approach, 612
 - live auto-tuning, 610
 - live musical performance, 626
 - model development
 - data preparation, 613
 - key model creation, 613–616
 - note probability and confidence, 616–619
 - musical note recognition, 621, 624
 - musical notes, 611
 - musical performance, 609
 - note distribution, 611
 - note usage, 611
 - note weighting, major and minor keys, 611, 612
 - performance, 626
 - pitch correction, 621–626
 - pitch stream, 621, 624
 - singer’s audio waveform, 620
 - singing, 609, 621
 - song collection, MusicXML format, 612
 - song note weights, 612
 - tonal analysis, 627
 - tonal center, 626
 - vibrato, 621
 - waveform data, 620
- Memory fragmentation, 499
- Message authentication, 520
- Message authentication code (MAC), 515, 517
- Message passing interface (MPI), 45
- Messages, 353
- Metacognition, 813, 815–818, 821
- Meta-instructions, 440, 441, 452
- Meta-knowledge, 200
- Microcontrollers
 - Arduino Uno, 202
 - M5 Stack, 205
 - NodeMCU ESP32, 202
 - NodeMCU ESP8266, 202
 - option comparisons, 202, 203
 - Particle Photon, 202
 - platform, 202
 - Raspberry Pi, 202
 - sensing applications, 202
- Microservice, 283, 284
- Migration diagram, 468
- Military defense coalitions, 545
- Military exercises and trials, 546

- Millennials, 298, 299
- Minimum pumping length (MIPU)
 - active learning tool, 143
 - features, 144
 - formal languages and automata, 147
 - GitHub repository, 159
 - goal, 143
 - main menu, 154
 - with major components, 148
 - membership testing functionality, 153
 - minimum pumping length determiner, 153–154
 - modules, 158
 - pumping lemma for regular languages, 146–147
 - regular expression, 155
- Minkowski transform, 473
- The missing theory, 473
- MIT App Inventor, 20
- Mixed-integer linear programming model (MILP)
 - assumptions, 536
 - container vessel, 533
 - CPLEX, 535, 542, 543
 - decision variables, 537
 - L-container loading process, 535, 539–542
 - loading process, 534
 - MIP, 535
 - notations, 536–537
 - optimization, 535
 - quay cranes, 534, 535
 - scheduling problem
 - quay cranes, 534, 535, 543
 - Tripoli-Lebanon port, 534
 - yard truck, 535, 543
 - storage location, 533, 541
 - Tripoli-Lebanon port, 534, 542–543
 - U-container unloading process, 534, 539, 542
 - yard trucks, 533, 538, 540, 543
- Mobile tools, 79
- Model-based systems engineering (MBSE)
 - agile practices, 744
 - and agile software development, 741, 744
 - diagrams and text artifacts, 743
 - document-based approach, 742
 - SysML, 743
 - systems, 741
- Modeling automotive embedded systems, 559–560
- Modeling systems' interaction, 459
- Modeling vehicle systems, 559
- Modelling and simulation, EDI
 - ABM, 267, 275
 - aggressive EDI interventions, 272
 - algorithmic approach, 269, 270
 - applicant pool, 271, 272
 - baseline assumptions, 267
 - best applicant, 272, 273
 - boilerplate university-wide EDI policy, 268
 - comparative analyses, 275
 - corner cases, 267–269
 - full EDI incentives, 269, 272
 - incentives/interventions, 267, 274
 - limitations, 275
 - statistics, 272, 274
- Modern dependability assurance, 512
- Modern paradigm
 - hardware/software cooperation, 438
 - omissions, 437
 - segregated computer components
 - misconception, 438
 - “sequential only” execution misconception, 438
- 15-Module multidisciplinary course
 - module 1, 45
 - module 2, 45
 - module 3, 45
 - module 4, 46
 - module 5, 46
 - module 6, 47
 - module 7, 47
 - module 8, 47
 - module 9, 47
 - module 10, 47
 - module 11, 47
 - module 12–14, 47
 - module 15, 47
- MongoDB data, 337, 344, 346
- Monte Carlo method, 46
- Moodle courses, 78, 219, 793, 794, 801, 805–807, 850, 853–857
- Motivational Theory of Role Modeling, 181
- MQTT (IoT frameworks), 114
- MQTT broker, 211
- M5Stack, 204
- M5Stack ESP32, 207
- Multi-Core/Many-Core (MC) processors, 436
- Multidimensional Software Engineering course, 88
- Multidisciplinary education, 49
- Multidisciplinary team, 53
- Multi-Intelligence All source Joint ISR Interoperability Coalition (MAJIIC 2), 554
- Multilayer Perceptron, 307, 308, 313
- Multimedia-based courses, 123
- Multimedia technologies, 17

Multiple representations, 23
 Musical note recognition, 621
 Musical notes, 621
 Music teachers
 experiences, 810
 experimental lessons, 817
 FD project, development (*see also* FD
 (university teacher professional
 training))
 activities, learnings and worksheets,
 815
 enhancing cognition and metacognition,
 815–817
 “Grassroots FD Project 2019”, 815
 TESOL, 810
 MusicXML format, 613
 Myers–Briggs Type Indicator (MBTI), 598
 MySQL database, 211
 MySQL Enterprise version, 357

N

National attrition, 4
 National Center of Academic Excellence
 in Cyber Defense in Four-
 Year Baccalaureate Education
 (CAE-CDE 4Y), 32
 National Council for Vocational Training
 (NCVT), 337
 National Household Education Surveys
 Program, 352
 National informatics (NIC) server, 344
 National Institute of Standards and Technology,
 642
 National Science Foundation (NSF), 43, 179
 National Security Agency (NSA), 32
 National Strategic Computing Initiative, 53
 Nearest Neighbor, 307–309
 Network Defense (ND), 33
 Networked edge devices, 472
 Network Forensics (NF) course, 33
 data collection and analysis, 36–37
 instructional units, 35
 instrumentation, 35
 participants, 34
 student demographics, 36
 Networking, 77
 Networking admin course auto-graders, 833
 Network-like addressing scheme, 450
 Network security certifications, 40, 41
 Neuromorphic architectures/applications, 451
 Neuromorphic systems, 487
 NIST 798-30 (risk assessment process), 513
 NIST 798-175B Cryptographic Standard, 514

NIST Cybersecurity Framework, 513
 Node deletion algorithm, 322
 Node.js, 283
 NodeMCU ESP32, 202, 204
 NodeMCU ESP8266, 202
 Nodes, 321–325
 Non-data visualization, 133
 Non-hardware concepts, 201
 Non-ICT students, 12
 Non-major CS lower-division coursework, 218
 Nontechnical/ professional focus, 8
 North Atlantic Treaty Organization (NATO),
 545
 NoSQL databases, 337, 346
 Note detection, 609, 610
 Note probability, 616–619
 Novice programmers, 9
 NSF program, 48
 NUS Computing Department, 81
 NUS Computing graduates, 73, 74

O

Object code, 452
 Object-oriented paradigm, 630
 Object oriented programming (OOP)
 ACEduSys, 847
 adaptive learning, 844, 847, 848
 BJP and BPP course, 849
 CW method, 855–857
 DPLP, 856
 learning technique, 847
 LMS, 850
 methodological meta-model, 848
 research goals, 848–849
 research questions, 848
 Object references, 499
 Objects, 496
 One-on-one interviews, 26
 Online auto-graded systems, 826
 Online course platforms
 e-learning, 781
 at large scale, 781, 782
 learner and instructor, 783
 learner satisfaction, 781, 782
 learners’ opinions (*see* Learner’s opinions
 in online learning platforms)
 primary and secondary category, 782–783
 recommendation, 781, 786, 790
 sentiment analysis, 781, 784, 786, 790
 Online cybersecurity graduate programs, 825
 Online Discussion Forum, 51
 Online learning, 838
 Online marketplace, 709

- Online tools, 79
 - Online trading, 962
 - Online training, 49–51
 - Online training creation
 - agenda and methodologies, 51
 - challenges, 52
 - choices, 49
 - face-to-face, 50
 - FC educational model, 50
 - foster communication skills, 49
 - fundamental goals, 49
 - HPC Facility, 50
 - includ software, 50
 - instruction and research, 51
 - 15-module program, 51
 - multidisciplinary team, 50
 - pedagogical techniques, 50
 - RA/TA, 51
 - synchronous meetings, 50
 - team-based homework, 50
 - Open content development (OCD), 260, 261
 - Open Dependability Exchange Metamodel (ODE), 513
 - Open-source software
 - algorithms, 634
 - Bad Practice or Correctness, 633
 - bug distribution rate, 634, 635
 - bugs, 632, 633
 - bug types, 635–637
 - database-related source code, 634
 - data structures, 634
 - Dodge Code category, 638
 - elementary programming source code, 634
 - groups, 634
 - quality of textbook code, 629, 630
 - research methods
 - bug data collection, 631–632
 - code analyzer tools, 631
 - static code analysis, 630, 631
 - teaching and learning programming, 629
 - texts group composition, 632
 - violation types, 637
 - Operating system (OS), 37, 77, 436
 - Operators on Gödel numbers, 421–424
 - Optimization, 156, 443, 507, 535, 786
 - Oral communication skills, 56, 57, 63
 - Organizing ‘ad hoc’ structures
 - code fragment, 447
 - fundamental cooperation method, 448
 - ‘hiring’ core, 446
 - parent-child relationships, 447
 - Oscillations, 257, 382, 622
 - Overall scheduling requirements, automotive cooperative development
 - AUTOSAR (*see* AUTOSAR)
 - composition technique, 559
 - genetic algorithm-based heuristic, 559
 - mapping description, 563–565
 - modeling automotive embedded systems, 559–560
 - port accesses, 565
 - PortChain*, 563, 565
 - repository organization, 560
 - transformation technique, 559
- P**
- PA intervention, 192
 - Pair programming, 219, 222
 - Parallelized sequential computing, 472
 - Parallelized sequential processing, 484
 - computational load, 486
 - computing objects, 486
 - computing performance, 487
 - computing systems, 485
 - cores, 486
 - distributed parallel processing, 485
 - HPL-class benchmarks, 486
 - idle time, 486
 - neuromorphic computing systems, 485
 - parallelization, 485
 - payload performance, 486
 - signal propagation, 485
 - SPA systems, 487
 - technical implementations, 486
 - Parent-child involvement, 351
 - parentOfNode*, 323, 324
 - Parent-school involvement, 351
 - Parent-teacher conferences, 352
 - Parent-teacher organizations, 352
 - Parent-Teacher Portal (PTP)
 - Bloomz*, 355–356
 - children, 353
 - ClassDojo*, 355
 - ClassTag*, 356
 - discussion board, 360
 - elementary school teachers and parents, 360
 - ELLs, 360
 - functional requirements, 358, 360
 - goal, 360
 - home-school partnership, 353
 - non-functional requirements, 358
 - parents, 354
 - Remind*, 354–355
 - student’s progress, 353
 - system implementation, 356–357
 - teachers, 354

- tools, 354
- usability, 358
- user interface, 358–361
- Parent-teacher relationship, 351, 353
- PA role model
 - comparison control course, 184–185
 - CSCI 549 (Intelligent Systems), 183, 184
 - diversity, 183
 - evaluation instruments, 185–186
 - evaluation plan, 187
 - in-class research projects, 184
 - intervention course, 185
 - official course description, 184
 - participants, 184
 - PhD students, 183
 - positive outcomes, 183
 - students' identification, 183, 184
 - theoretical model, 183
- Partial differential equations (PDEs), 45
- Particle Photon, 202, 203
- Passive control mechanisms, 31
- Passive infrared (PIR) sensor, 209
- Patrol system, 944–945
- Pearson's Correlation Matrix, 125, 127
- Pedagogical approach, 17
- Pedagogical tools, 279
- Peer assistant (PA), 186, 189
- Peer instruction/pair programming, 218
- Peer-review component, 199
- Peer-to-peer (P2P) network, 169
- PeopleNTech (IT professional skills development institute)
 - BRBES approach, 233
 - certificate/diploma, 236
 - classroom template, 234
 - curriculum, 232
 - essential instructional elements, 234–235
 - evaluation
 - assignments/labs/quizzes, 236
 - class tests, 236
 - post-course boot camp lab, 236
 - student test preparation assessment tools, 236
 - vendor exam preparation, 236
 - 4-month program, mid-level/senior-level IT
 - employment, 232, 233
 - industry professionals, 233
 - industry trends, 233
 - IT training model, 244, 245
 - learning component identification, 233
 - post-class survey, 237
 - priority, 233
 - research latest industry trends, 234
 - students, 232
 - top-flight job placement support
 - job placement services, 237–238
 - mock interview sessions, 237
 - resume assistance, 237
- Percentage relevance, 75
- Performance evaluation, 494, 501
- Peripherals, 205
- Personal Competence Domain Model (PCDM), 849
- Personality, 304
- Personal protective equipment (PPE), 106, 107, 266
- Personal statement, 48
- PhD student researchers, 182
- Physical manual therapy, 279
- Physical therapy
 - digital tool, 278
 - manual therapy, 279
 - professionals, 278
 - quantitative metrics, 278
- Physical therapy analytics dashboard
 - API, 279
 - dashboard development
 - back end, 283–284
 - deployment, 285
 - front end, 281–282
 - desktop application, 279
 - operating systems, 279
 - pedagogical tool, 279
 - pressure sensing fabric technologies, 285
 - Studio 1 Labs sensor fabric, 280, 285
 - users, 279
- Pitch correction, 621–623, 625–626
- Platform-as-a-service cloud solution, 709
- Platform-based applications, 71
- Platoon use case
 - asymmetric vs. symmetric encryption, 517
 - bidirectional with broadcast message
 - on-the-fly certification, 518
 - with pre-certification, 518
 - bidirectional with on-the-fly certification, 517–518
 - communication model, 516
 - on-the-fly systems, 516
- 4-Point Likert scale, 35
- Pollard's rho method, 395
- Polymorphism, 495
- Portal, 357
 - KM-EP, 793, 800
 - Microsoft Azure portal, 170
 - PTP (*see* Parent-Teacher Portal (PTP))
 - web-base portal, 353

- PortChain*, 563–565
- Postcard assignment assessment
 analysis/synthesis, 138
 Bloom’s cognitive, 138
 data literacy, 138
 electronic documents, 137
 information literacy, 138
 instructional goals, 136
 learning outcomes, 136
 mode of communication, 137
 statistical indicators, 137
 students’ perception, 136
 survey instrument, 136
- Postgraduate program, 81
- PostgreSQL relational database, 284
- Post-matric scholarship (PMS) scheme, 346
- Practice run, 114
- Pre-and post-data test, 26, 27, 42
- Precision
 and accuracy, 280, 285
 bounded floating point, 367
 definition, 365
 floating-point format, 368
 precisely zero, 369, 370
- Pre-employment curricula, 230
- Pre-employment education, 231
- Prefrontal cortex, 304
- Presentation tools, 78
- Pressure controller’s protocol, 467
- Pressure monitor’s protocol, 466
- Pressure sensing fabric technologies, 285
- Pressure sensor’s protocol, 466
- Pre-survey responses, 223
- Priority queue, 506
- Prior programming experience, 9
- Problem-solving skills, 64
- Processing time, 474, 475
- Processing unit (PU), 437–439, 471
- Product knowledge, 525
- Professional skills, 6, 56
- Program accreditation requirements, 103
- Programming auto-graders
 derivatives
 of advanced programming labs, 832
 of expression labs, 832
 test cases, 831
- Programming languages, 45, 80
- Proof-of-work (PoW) algorithms, 164, 174
- Proper sequencing, 455
- Prototypes, 110
- Pseudo-random quantum circuits, 412, 413
- Public Key Infrastructure (PKI) certification, 517
- Publish/subscribe system
 content-based, 910–912
 efficient fuzzy matching algorithm, 914–915
 fuzzy matching algorithm, 913–914
 logical coverage relationship between constraints, 915–916
 map-based, 912
 matching algorithms, 910
 subscription constraint value, 917
 subscription information, 916–917
 theme-based, 910
 XML-based, 912
- Pumping lemma
 and minimum pumping length, 144
 MIPU educational software (*see* Minimum pumping length (MIPU))
 for regular languages, 143, 144, 146–147
 LSG, 153
 MIPU, 153–154, 158–159
 NFA class, 151–153
 regular expression to NFA converter, 148–151
- Purpose-build HW brain simulator, 481
- Python, 283, 824, 831, 833, 834, 849, 901, 903, 906, 907
- Python-based *music21* toolkit, 613
- Python code, 614
- Q**
- QBL Plugin for Moodle (QBLM4Moodle), 850, 854, 856, 857
- QRT maps, 398–399
- Qualifications-based learning model (QBLM), 794–797, 849, 850, 854
- Quantitative analysis, 57
- Quantitative and qualitative methods, 36, 42
- Quantum circuits, 412, 413
- Quantum random
 quantum random Bernoulli trials, 377
 random instruction, 375, 377
- Quantum supremacy, 411–412, 414–417
- Quasi-experimental design, 140, 185
- Quasi-thread (QT), 439, 446
- Quay crane and yard truck scheduling
 problems (QCYTSP), 534, 535, 543
- Quay cranes, 533–543
- Queue, 506–507
- R**
- “Rabbit-hole,” YouTube’s algorithm, 775–778
- Race Against Time module
 app simulation, 21

- arrow diagrams, 22, 27
 - linear equations, 19
 - linear transformations, 23
 - relay races, 19
 - Relay Race simulation, 22
 - stack cubes, 22
 - Radiative transfer simulation framework, 46
 - Radicalization, 775, 776, 778
 - Random Forest, 307, 308, 311, 317
 - Random Tree, 307, 311–313
 - Raspberry Pi, 202, 204
 - Rational unified process (RUP), 938
 - Reading, 837, 838, 841, 843
 - Reallocation, 497
 - Real numbers, 365
 - Real-time component-based systems, 463
 - Real time operating systems (RTOS), 205
 - Real-time systems, 557
 - Receiver operating characteristics curves (ROCs), 243, 244
 - Recruitment, 48
 - Reduced power consumption, 453
 - Redundancy, 453
 - Register-to-register transfer, 453
 - Regular language
 - active learning, 144
 - and finite automata, 143
 - pumping lemma, 143, 144, 146–147, 154–159
 - regular expression, 144
 - to NFA converter, 148–151
 - Relative location algorithm, 321–325
 - Relay races, 19, 20
 - Relevance
 - computer programming, 80
 - databases, 79
 - desktop publishing, 78
 - graphics, 79
 - hardware, 77
 - improvement recommendations, 75–77
 - information systems, 80
 - information systems management, 80
 - mobile tools, 79
 - networking, 77
 - online tools, 79
 - operating systems, 77
 - percentage relevance, 75
 - presentation tools, 78
 - spreadsheets, 78
 - technology usage frequency, 75, 76
 - word-processing, 78
 - Remind* app., 354–355
 - Removal operations, 506
 - removeAll method, 504
 - removeFirst method, 505
 - Repository organization, 560
 - Representational logic, 27
 - Representational State Transfer (REST), 197, 283, 570, 796, 801
 - Research identity, 183
 - Resilient distributed datasets (RDD), 47
 - Resource sharing without scheduling, 454
 - Risk assessment process, 513
 - Risk detection, 340
 - “Risk probability chart”, 937
 - Risks analysis, contingency plan, 937, 938
 - Road Coloring app simulation, 24
 - Road Coloring module
 - App Inventor, 21
 - challenge based, 19
 - constructing cities, 20
 - mathematics research, 19
 - mobile app simulation, 20
 - Red and Blue functions, 26
 - simultaneous physical movements, 19
 - synchronizing instructions, 20, 21
 - visual programming block system, 20
 - Robot Operating System (ROS), 514
 - Role models, 181
 - ROS messages, 514
 - RSA certification, 517
 - RSA encryption/decryption, 211
- S**
- Safety-critical IoT systems
 - behavior specification methods, 459
 - ETBP (*see* Enhanced time behavior protocol (ETBP))
 - formal modeling methods, 459
 - high quality, 459
 - interconnection structure, 460
 - specification language, 460–463
 - Salesforce, 706, 709–712, 716
 - Satabase platforms, 81
 - Scalar multiplication, 397, 405–406
 - Scheduling problem, 534, 535, 543
 - Scholarship schemes, 338
 - School of Computing and Information Technology (SCIT), 123
 - secBIML programming language, 826
 - Second introductory programming unit, 9
 - Secure Socket Layer (SSL), 515
 - Security, 358
 - Security control, 514
 - Security countermeasures, 38
 - SEED project, 33
 - Segregated processors, 472

- Selection criteria, 48
- Self-concepts, 180
- Self-driven cars, 867, 869, 873, 875, 943–948
- Self-driving/monitoring system
 - central control unit, 943
 - cost, budget and suppliers, 946
 - delivery system, 944
 - food delivery system, 945
 - low-speed self-driving cars, 943–945
 - market analysis
 - marketing assets, 948
 - target customers, 947–948
 - mission, 943
 - patrol system, 944–945
 - Starship Technologies, 943
 - SWOT analysis
 - internal strengths, 946
 - internal weaknesses, 946
 - opportunities, 947
 - outside threats, 947
 - system, self-driving cars, 944
 - technologies, 945–946
 - vision, 943
- Self-efficacy, 181, 189
- Self-modification, 375, 389
- Self-paced learning, 8
- Self-Reflective Journaling (SRJ), 292–293
- Sensor fabric, 280, 283
- Sensors, 200, 204–205, 207
- Sentiment analysis
 - product reviews on social media (Twitter), 899–906
 - Python script, 901
- Sequential programming model, 435
- Serve as a medical device (SaMD), 643
- Set cover problem, 963–964
- Shared temporal cognitions (STC), 526, 528
- Short snappers, 108
- Short vowel dyslexia (SVD), 838, 841–843
- “Similar numbers”, 369
- SIMnet (commercial e-learning product), 827
- Simulated pseudo-random quantum circuits, 412, 413
- SIMULINK program, 260
- Singing, 609, 621
- Single Processor Approach (SPA), 435, 451
- Single-processor performance, 471
- Skill development, 7
- Skills competency assessment, 230
- Skills Framework for the Information Age (SFIA), 7
- Skip controller, 110, 111, 113
- Skype, 79, 949–955, 957, 958
- Slack instant messaging (IM), 52
- Sleep, 303, 309
- Sleep deprivation, 304
- Smart contracts, 648
- Social capital integration and technological integration (SIST), 873
- Social dynamics, 114
- Social integration, 8
- Social justice schemes, 338
- Social media, 117, 118, 304, 312, 886
 - in higher education, 119–120
 - sentiment analysis, product reviews, 899–906
 - TextBlob, 903–906
 - valuable data, 900
- Social network sites, 117, 120
- Social welfare schemes, 338
- Soft skills, 81
- Software Development Life Cycle (SDLC), 653
- Software Development (SD) major, 7
- Software engineering (SWE)
 - graduates, 88
 - SWEBOK-V3.0, 85, 86, 747, 748
 - SWE-KAs (*see* Software Engineering Knowledge Areas (SWE-KAs))
 - for web applications, 748
- Software Engineering Knowledge Areas (SWE-KAs)
 - coverage (*see* SWE-KAs coverage)
 - groups, 89
 - research approach, 749
 - software construction, 749, 757–758, 762–766
 - software design, 748–749, 755, 756, 762, 764–766
 - software maintenance, 749, 761, 763–766
 - software requirements, 747–749, 752–754, 762, 764–766
 - software testing, 748, 749, 752, 759, 760, 762, 763
 - SWEBOK-V3.0, 86, 747
 - SWE-Courses, 85
 - SWE-Curriculum, 87
 - SWE-KA#6, 86
 - SWE-KA#7, 86
 - SWE-KA#8, 86
 - SWE-KA#9, 87
 - SWE-KA#10, 87
- Software Engineering Undergraduate Program (SWE-Curriculum), 85
- Software engineers, 72
- Software evaluation
 - assessment methods, 891
 - automotive industry, 887

- cost-oriented methods, 881–882, 889–890
- cross-industry use, 886
- digital goods, 880
- empirical study, 882, 891, 895
- explanatory sequential design, 883
- license-oriented methods, 882, 890
- quantitative survey, 884–885
- questionnaire design, 885–886, 891
- research goal and design, 882–883
- value-oriented methods, 882, 890
- Software evolution, 671
- Software implementation, 494
- Software process tailoring (SPT)
 - case research method, 529
 - collaborative activity, 524
 - conflicts, 524, 525
 - coordination and time-based compatible behavioral patterns, 528
 - exchange, 529
 - impacts, 524
 - knowledge-and learning-intensive activity, 523, 525
 - process knowledge, 525
 - process modification strategies, 525
 - product knowledge, 525
 - project goal assessment, 524
 - software development, 524, 525, 527
 - software development project, 523
 - statistical techniques, 529
 - STC, 526, 528
 - tailoring decisions, 527
 - tailoring strategies, 525
 - teams' operations vs. behaviors, 523
 - theoretical model, 528
 - TMS, 523–528
 - types of knowledge, 526
 - validation and evaluation, 525
- Software procurement
 - cost-oriented methods, 881–882, 889–890
 - digital transformation, 879–880
 - license-oriented methods, 882, 890
 - MR, 974, 979–981
 - quantitative survey, 884–885
 - value-oriented methods, 882, 890
- Software Quality Assurance, 87, 94–96
- Software Reliability Growth Models (SRGM), 671–675, 682–685
- Software systems, 671
- Software tests, 546
- Solar energy
 - technologies, 729–730
 - and wind energy, 729, 731
- Solar power, 729
- Somos sequences, 397, 399–401
- Space complexity, 493, 498
- Spark, 47
- 'Sparse' calculations, 453
- Spreadsheets, 78
- Stack, 495, 505–506
- STANAG 4559 standard
 - AEDP-17, 548
 - CSD, 547
 - CSD-Server, 548
 - CSD-Streaming Server, 548
 - implementations, 547
 - IRM&CM, 547
 - JISR process, 547, 554
 - JISR Test Center, 550
 - military surveillance, 547
 - products, 547
 - reconnaissance environment, 547
 - SOS, 547
 - testing
 - CORBA interface, 548
 - web service interfaces, 549
- Standard floating-point format, 365
- Standardization Agreements (STANAGs), 545
- Stanford's model of design thinking, 107
- Starship Technologies, 943
- Static scheduling, 558
- Static structure, 561
- Statistical indicators, 137
- STEM education research, 185
- STEM fields, 179, 181
- STEM majors, 217
- Stereotype Inoculation Model, 181
- Streaming models
 - Android, 661
 - app interface, 658
 - apps class structure, 668
 - architecture, 657
 - audio focus, 661
 - background process, 658–659
 - communication channel, 659
 - components, 658
 - content player, 659
 - data storage, 660
 - developers, 657
 - framework class structure, 661
 - framework implementation
 - Android devices, 662
 - app interface, 663
 - app new features, 664
 - audio cable/Bluetooth, 662
 - Broad-castReceiver*, 663
 - MainActivity, 663, 664
 - MediaPlayer, 665–666
 - message broadcast receiver, 665

- Streaming models (*cont.*)
 - multimedia content, 662
 - service, 665
 - validator, 662
- generic framework, 657
- mobile-connected devices, 657
- mobile devices, 657
- network lock, 660
- power manager, 659
- power manager and wake lock, 666
- threads, 660
- URLs, 660
- user permissions, 661, 666–668
- Strengths, weaknesses, opportunities and threats (SWOT)
 - self-driving system
 - internal strengths, 946
 - internal weaknesses, 946
 - opportunities, 947
 - outside threats, 947
- STRIDE (threat modeling technique), 513
- Structural templates (ST), 794
- Student-content interaction (SC), 119, 122, 125, 128
- Student misconceptions, 4, 6, 7
- Students' receptiveness, 114
- Student-student (SS) interaction, 8, 119, 122, 125, 128
- Student-teacher (ST) interaction, 119, 122, 125, 128
- Studio 1 Labs sensor fabric, 280
- Subroutine call without stack, 454
- Supercomputer Aurora'18: "*Knights Hill*", 437
- Supercomputers, 436
- Supercomputing, 412, 413
- Survey (RINCCIII), 69
- Sustainable design, 105
- Sustainable development, 105
- SWEBOK-V3.0, 85
- SWE-Curriculum at JUST
 - challenges, 88
 - compliance, 95, 96
 - IET, 85
 - problem-based learning approach, 87
 - Software Quality Assurance*, 96
 - surveyed SWE, 88
 - SWE courses, 86
 - SWE-KAs (*see* Software Engineering Knowledge Areas (SWE-KAs))
- Sweepers, 110
- Sweeping, 110
- SWE-KA#6 (Software Configuration Management), 86, 90–91, 96
- SWE-KA#7 (Software Engineering Management), 86, 91–93, 96
- SWE-KA#8 (Software Engineering Process), 86, 93–94
- SWE-KA#9 (SWE Models and Methods), 87, 94–96
- SWE-KA#10 (Software Quality), 87, 94
- SWE-KAs coverage
 - classification, CLOs, 89–90
 - inspection, 89
 - SWE-KA#6 (Software Configuration Management), 90–91
 - SWE-KA#7 (Software Engineering Management), 91–93
 - SWE-KA#8 (Software Engineering Process), 93–94
 - SWE-KA#9 (SWE Models and Methods), 94–96
 - SWE-KA#10 (Software Quality), 94
- Symmetric cryptographic encryption, 519
- Symmetric key encryption, 517
- Symmetric keys, 519
- Synchronization, 445, 446
- Synchronization tests, 552–553
- Synchronize metadata, 552–553
- Synchronous computing, 480
- SYSBOOK platform
 - case studies, 253
 - categories, 253
 - cover page, 252
 - demonstration, 253
 - Java applet, 253, 254
 - knowledge, 251
 - multilevel e-book, 251
 - OCD model, 260, 261
 - PID controller, 254, 255
 - system behavior, 253
 - system control, 253
 - time vs. frequency domain, 254
- Systematic literature review (SLR), 642, 644, 653
 - blockchain-oriented solutions, 648
 - data extraction, 647
 - ensuring interoperability, 651–653
 - ERP systems
 - with big data, 867–868
 - big data with automotive industry, 868–869
 - complexity, ERP responsiveness, 871
 - data management, 869
 - implementation, 866
 - influence on automotive industry, 868
 - research gaps, 869, 870
 - statistics, 864

- trust issues, 871
 - evaluation process, 646
 - meeting regulatory requirements, 648, 650
 - planning, 645
 - process, 644
 - publication years, 647
 - quality assessment, 646, 647
 - security and protection of privacy, 650–651
 - selection, 646
 - System of Systems (SOS), 547
 - Systems engineering artifacts, 742
 - System's integrity, 514
 - System's model-based safety reflection, 512
 - Systems Modeling Language (SysML), 743
- T**
- Tableau Public training data, 135
 - TADL metamodel, 559
 - Tamper-resistant characteristics, 648
 - Tasmanian ICT domestic commencements, 12
 - Tasmanian ICT sector, 5
 - Teaching, 278, 291, 293, 300
 - Teaching assistant (TA), 186, 189, 825
 - Team-based frontier research projects, 44
 - Team-based with participants, 44
 - Technical debt categorization
 - CRM, 715, 716
 - design and people type categories, 713, 714
 - entries, 714
 - granular classification, 715
 - intention level, 707, 710–711, 713, 715
 - level, 716
 - matching entries, 712
 - nature, 708, 709
 - OrgConfessions, 710
 - people type technical debt, 714, 715
 - process-related debts, 715
 - research, 716
 - software development process, 711, 716
 - technical debt quadrant, 707–708, 711
 - Technical debt, CRM application
 - awareness, 706
 - categorization (*see* Technical debt categorization)
 - code defects, 707
 - enterprise-level software, 705, 709
 - identification methods, 707
 - ISV, 706
 - phases, 710
 - research methodology, 710
 - salesforce, 706, 709–710, 714, 716
 - software development, 705
 - types, 708
 - validation process, 711–713
 - Ward Cunningham, 707
 - Technical debt quadrant, 707–708, 711
 - Technical educational data, 341
 - Technical skill competence, 64
 - Technical skills, 6, 229
 - Technologies
 - solar energy, 729–730
 - Starship, 943
 - Technology Modernization Action Plan*, 641
 - Temporal behavior effects, scaling, 488
 - Temporal logic
 - 1-bit adder, 476–478
 - computing chain effect/technology/material, 478–480
 - 3-dimensional coordinate system, 474
 - 4 dimensional time-space system, 473
 - hypothetical experiment, 474
 - temporal behaviour, 475–476
 - TESOL (Teaching English to Speakers of Other Languages), 810
 - Test-driven development (TDD), 546
 - Testing, JISR Test Center
 - A.C.T.S. test cases, 551, 553
 - AEDP-17, 550, 554
 - API, 551
 - CORBA interface, 551
 - requirements, 549–550
 - STANAG 4559, 550, 554
 - synchronization tests, 552–553
 - updatemethod, 551
 - Theory of recursive functions, 421–433
 - Thermostat readings, 201
 - The von Neumann architecture*, 426
 - Third party software (TPS)
 - in cloud
 - compliance and cybersecurity risk, 770
 - problem, motivation and importance, 770
 - third-party tools, 769
 - compliance, 771, 772
 - coronavirus pandemic, 772
 - investment for companies, 769
 - literature reviews, 772
 - security, 770–771, 773
 - Threat and risk analyses (TARA), 512
 - 3D additive manufacturing, 111
 - Three-dimensional approach, 25
 - “Three domains of learning”, 230
 - Tic Tac Toe, 421
 - Time complexity, 501
 - Timed automata, 459
 - Time-space system, 480

- Time transformation (TT), 672, 675, 678, 679, 683, 686, 690, 691
- TM4C123GXL LaunchPad Development Kit, 109
- Tonal center, 616, 626
- Total displacement, 22, 23
- Tourism in Saudi Arabia, 961–962, 964
- Tourism market
 - clearinghouse, 963
 - consortium, 962–963
 - electronic market, 961, 962, 964
 - e-marketplaces, 962
 - market model, 965
 - proposed electronic market, 964
 - in Saudi Arabia, 961–962
- Tourists
 - activities lists and service matchings, 966, 968
 - AIUla, 961
 - automated, electronic and agent-based auction, 965
 - in Saudi Arabia, 961–962
 - and the service provider, 962
- Toward a Theory of Instruction*, 24
- Traditional computer science undergraduate programs, 41
- Traditional design process criteria, 108
- TrainDys system, 838, 840, 841, 843
- Training-based Workforce Development for Advanced Cyberinfrastructure (CyberTraining), 43
- “Training Data” themes, 135, 141
- Transactive memory system (TMS)
 - shared system, 525
 - software teams, 524
 - SPT knowledge exchange, 528
 - SPT performance, 524, 527, 528
 - SPT process, 525–527
 - team members, 525
- Transformation technique, 559
- Transit between system components, 515
- Transit from system to system, 516
- Transmission speed, 475
- Transmission system operators (TSOs), 569
- Transparent Data Encryption (TDE), 357
- Transport Layer Security (TLS), 515
- Transport modes, 450
- Tree applications, 320–321
- Trees, 319
- Turing machine
 - autonomous dynamical system, 390–392
 - computation, 375, 376
 - halting problem, 375, 388–389
 - incomputable languages, 382, 387
 - self-modifying programs, 376
 - standard machine, 377
- Tutorial exercises and assignments, 8
- Twitter
 - APIs, 902–903
 - Flask, 906
 - positive, negative and neutral tweets, 900
 - sentiment analysis, products
 - results, sentiment analysis, 901
 - system architecture design, 901
 - TextBlob, 903–906
 - Tweepy, 903
 - tweets, 900–902
- Two-tailed paired t-test, 37
- U**
- Ubiquitous Arduino Uno, 202
- Ubiquitous computing, 437
- UMBC CyberTraining initiative, 43
- Unified Modeling Language (UML), 851–855
- Unit cost depression, 880
- United Nation’s Sustainable Goals (UN SDG), 291
- Universities, 230
- University Grants Commission (UGC), 337
- University of Houston–Clear Lake (UHCL), 32
- University of Tasmania (UTAS), 3, 5, 13
- Unmanned aircraft system (UAS), 742
- Unpaired t-test, 59
- UN Sustainable Development Goals, 105, 287
- Upper-division CS courses, 219
- Upper-level traditional CS students, 41
- U.S.-based IT professional skills development institute, 232
- User interface, 358–361
- UTAS ICT courses, 5
- V**
- Verbal and written communication, 71
- VeriSM (proffered agile ITSM frameworks), 922, 925–927, 930–934
- Virtual processors, 439
- Visual programming block system, 20, 26
- Visual reasoning, 27
- Voice-controlled instrument, 622
- von Neumann architecture, 473, 488, 489
- ‘von Neumann’ bottleneck, 453
- Vowel dyslexia (VD), 838, 841–843
- VueJS, 282
- Vuex, 281

W

- Water auction market for Jazan
 - anticipated costs, water transportation, 733, 734
 - AVKO 365K model, 731
 - cost of water transportation, 733
 - desalination plant, 731
 - goal and constraints, 732–733
 - process, 734
 - sellers and buyers, 731
 - simulation result, 735–739
 - simulation system, 735
 - water from air, 732, 738
 - water market structure, 731, 732
- Wayfinding, 114
- Web application firewall (WAF), 520
- Web-based languages, 81
- Web-based text processing tools, 297
- Web-base portal, 353
- WebEx Teams, 52
- Web service interfaces, 549
- Weighted end-of-unit exam, 9
- Welch's t-test, 59, 61, 62
- 'Wet' neuro-biology, 473
- WhatsApp, 117, 118
- WhatsApp groups usage, research
 - academic purposes, 123
 - collaborative learning environment, 123
 - contributions, 118–119
 - data collection instrument, 124
 - data preparation, 125
 - descriptive analysis, 125–126
 - findings, 126
 - higher education, 120–121
 - hypothesis, 119
 - instructors, 118
 - limitations, 119, 125
 - LMS, 118, 120
 - population and sample, 123–124
 - research design, 123
 - research questions, 119, 126–128
 - sample course description, 124
 - social class setting, 117
 - theoretical framework, 121–122
 - variables, 119
- WhatsApp's nonintrusive social media setting, 128
- Wide Area Network (WAN), 548
- Wi-Fi communication, 209
- Wind energy
 - AWG, 730
 - definition, 730

- electricity generation, 730
 - renewable energy, 730
 - water pumping, 730
- Wireless communication, 110
- Wireless dynamic network, 911
- Word Cloud, 296
- Word-processing skills, 78
- Work/study/internship variety, 71
- World Economic Forum, 81
- World Wide Web, 75
- Worst-case, 480
- Written communication skills, 56, 57

Y

- Yard truck, 533–543
- YOULA parameterization
 - basic control course, 255
 - control algorithms, 258, 261
 - controller *C*, 255
 - controller design, 258, 261
 - control stable processes, 256
 - control structure, 255, 256
 - filters, 255, 257
 - IMC, 256
 - negative feedback, 255
 - oscillations, 257
 - PID* controller, 255
 - process model *P*, 256
 - SIMULINK program, 259
- YouTube
 - CON group, pre-survey responses, 223
 - FC students, 219
 - Google, 776
 - pre-survey responses, CON group, 223
 - "rabbit-hole" algorithm, 775–778
 - tags, 775
 - US users, 776, 777

Z

- Zoom
 - collaboration tool, 949, 950, 956
 - cybersecurity, 956–957
 - GoToMeeting, 957
 - HIPAA, 49
 - online options, 958
 - online training, 49
 - privacy policy, 956
 - UC Berkeley's, 955–956
- Zuse Z4 computer, 365