

International Journal on

Advances in Software



2018 vol. 11 nr. 3&4

The *International Journal on Advances in Software* is published by IARIA.

ISSN: 1942-2628

journals site: <http://www.iariajournals.org>

contact: petre@iaria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Software, issn 1942-2628
vol. 11, no. 3 & 4, year 2018, <http://www.iariajournals.org/software/>

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Software, issn 1942-2628
vol. 11, no. 3 & 4, year 2018,<start page>:<end page> , <http://www.iariajournals.org/software/>

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA

www.iaria.org

Copyright © 2018 IARIA

Editor-in-Chief

Luigi Lavazza, Università dell'Insubria - Varese, Italy

Editorial Advisory Board

Hermann Kaindl, TU-Wien, Austria

Herwig Mannaert, University of Antwerp, Belgium

Subject-Expert Associated Editors

Sanjay Bhulai, Vrije Universiteit Amsterdam, the Netherlands (DATA ANALYTICS)

Stephen Clyde, Utah State University, USA (SOFTENG + ICSEA)

Emanuele Covino, Università degli Studi di Bari Aldo Moro, Italy (COMPUTATION TOOLS)

Robert (Bob) Duncan, University of Aberdeen, UK (ICCGI & CLOUD COMPUTING)

Venkat Naidu Gudivada, East Carolina University, USA (ALLDATA)

Andreas Hausotter, Hochschule Hannover - University of Applied Sciences and Arts, Germany (SERVICE COMPUTATION)

Sergio Ilarri, University of Zaragoza, Spain (DBKDA + FUTURE COMPUTING)

Christopher Ireland, The Open University, UK (FASSI + VALID + SIMUL)

Alex Mirnig, University of Salzburg, Austria (CONTENT + PATTERNS)

Jaehyun Park, Incheon National University (INU), South Korea (ACHI)

Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance (HLRN), Germany (GEOProcessing + ADVCOMP + INFOCOMP)

Markus Ullmann, Federal Office for Information Security / University of Applied Sciences Bonn-Rhine-Sieg, Germany (VEHICULAR + MOBILITY)

Editorial Board

Witold Abramowicz, The Poznan University of Economics, Poland

Abdelkader Adla, University of Oran, Algeria

Syed Nadeem Ahsan, Technical University Graz, Austria / Iqra University, Pakistan

Marc Aiguier, École Centrale Paris, France

Rajendra Akerkar, Western Norway Research Institute, Norway

Zaher Al Aghbari, University of Sharjah, UAE

Riccardo Albertoni, Istituto per la Matematica Applicata e Tecnologie Informatiche "Enrico Magenes" Consiglio Nazionale delle Ricerche, (IMATI-CNR), Italy / Universidad Politécnica de Madrid, Spain

Ahmed Al-Moayed, Hochschule Furtwangen University, Germany

Giner Alor Hernández, Instituto Tecnológico de Orizaba, México

Zakarya Alzamil, King Saud University, Saudi Arabia

Frederic Amblard, IRIT - Université Toulouse 1, France

Vincenzo Ambriola, Università di Pisa, Italy

Andreas S. Andreou, Cyprus University of Technology - Limassol, Cyprus
Annalisa Appice, Università degli Studi di Bari Aldo Moro, Italy
Philip Azariadis, University of the Aegean, Greece
Thierry Badard, Université Laval, Canada
Muneera Bano, International Islamic University - Islamabad, Pakistan
Fabian Barbato, Technology University ORT, Montevideo, Uruguay
Peter Baumann, Jacobs University Bremen / Rasdaman GmbH Bremen, Germany
Gabriele Bavota, University of Salerno, Italy
Grigorios N. Beligiannis, University of Western Greece, Greece
Nouredine Belkhatir, University of Grenoble, France
Jorge Bernardino, ISEC - Institute Polytechnic of Coimbra, Portugal
Rudolf Berrendorf, Bonn-Rhein-Sieg University of Applied Sciences - Sankt Augustin, Germany
Ateet Bhalla, Independent Consultant, India
Fernando Boronat Seguí, Universidad Politecnica de Valencia, Spain
Pierre Borne, Ecole Centrale de Lille, France
Farid Bourennani, University of Ontario Institute of Technology (UOIT), Canada
Narhimene Boustia, Saad Dahlab University - Blida, Algeria
Hongyu Pei Breivold, ABB Corporate Research, Sweden
Carsten Brockmann, Universität Potsdam, Germany
Antonio Bucchiarone, Fondazione Bruno Kessler, Italy
Georg Buchgeher, Software Competence Center Hagenberg GmbH, Austria
Dumitru Burdescu, University of Craiova, Romania
Martine Cadot, University of Nancy / LORIA, France
Isabel Candal-Vicente, Universidad del Este, Puerto Rico
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Jose Carlos Metrolho, Polytechnic Institute of Castelo Branco, Portugal
Alain Casali, Aix-Marseille University, France
Yaser Chaaban, Leibniz University of Hanover, Germany
Savvas A. Chatzichristofis, Democritus University of Thrace, Greece
Antonin Chazalet, Orange, France
Jiann-Liang Chen, National Dong Hwa University, China
Shiping Chen, CSIRO ICT Centre, Australia
Wen-Shiung Chen, National Chi Nan University, Taiwan
Zhe Chen, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China
PR
Po-Hsun Cheng, National Kaohsiung Normal University, Taiwan
Yoonsik Cheon, The University of Texas at El Paso, USA
Lau Cheuk Lung, INE/UFSC, Brazil
Robert Chew, Lien Centre for Social Innovation, Singapore
Andrew Connor, Auckland University of Technology, New Zealand
Rebeca Cortázar, University of Deusto, Spain
Noël Crespi, Institut Telecom, Telecom SudParis, France
Carlos E. Cuesta, Rey Juan Carlos University, Spain
Duilio Curcio, University of Calabria, Italy
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Paulo Asterio de Castro Guerra, Tapijara Programação de Sistemas Ltda. - Lambari, Brazil

Cláudio de Souza Baptista, University of Campina Grande, Brazil
Maria del Pilar Angeles, Universidad Nacional Autónoma de México, México
Rafael del Vado Vírveda, Universidad Complutense de Madrid, Spain
Giovanni Denaro, University of Milano-Bicocca, Italy
Nirmit Desai, IBM Research, India
Vincenzo Deufemia, Università di Salerno, Italy
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Javier Diaz, Rutgers University, USA
Nicholas John Dingle, University of Manchester, UK
Roland Dodd, CQUniversity, Australia
Aijuan Dong, Hood College, USA
Suzana Dragicevic, Simon Fraser University- Burnaby, Canada
Cédric du Mouza, CNAM, France
Ann Dunkin, Palo Alto Unified School District, USA
Jana Dvorakova, Comenius University, Slovakia
Hans-Dieter Ehrich, Technische Universität Braunschweig, Germany
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Atilla Elçi, Aksaray University, Turkey
Khaled El-Fakih, American University of Sharjah, UAE
Gledson Elias, Federal University of Paraíba, Brazil
Sameh Elnikety, Microsoft Research, USA
Fausto Fasano, University of Molise, Italy
Michael Felderer, University of Innsbruck, Austria
João M. Fernandes, Universidade de Minho, Portugal
Luis Fernandez-Sanz, University of de Alcala, Spain
Felipe Ferraz, C.E.S.A.R, Brazil
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Wolfgang Fohl, Hamburg University, Germany
Simon Fong, University of Macau, Macau SAR
Gianluca Franchino, Scuola Superiore Sant'Anna, Pisa, Italy
Naoki Fukuta, Shizuoka University, Japan
Martin Gaedke, Chemnitz University of Technology, Germany
Félix J. García Clemente, University of Murcia, Spain
José García-Fanjul, University of Oviedo, Spain
Felipe Garcia-Sanchez, Universidad Politecnica de Cartagena (UPCT), Spain
Michael Gebhart, Gebhart Quality Analysis (QA) 82, Germany
Tejas R. Gandhi, Virtua Health-Marlton, USA
Andrea Giachetti, Università degli Studi di Verona, Italy
Afzal Godil, National Institute of Standards and Technology, USA
Luis Gomes, Universidade Nova Lisboa, Portugal
Diego Gonzalez Aguilera, University of Salamanca - Avila, Spain
Pascual Gonzalez, University of Castilla-La Mancha, Spain
Björn Gottfried, University of Bremen, Germany
Victor Govindaswamy, Texas A&M University, USA
Gregor Grambow, AristaFlow GmbH, Germany
Carlos Granell, European Commission / Joint Research Centre, Italy

Christoph Grimm, University of Kaiserslautern, Austria
Michael Grottke, University of Erlangen-Nuernberg, Germany
Vic Grout, Glyndwr University, UK
Ensar Gul, Marmara University, Turkey
Richard Gunstone, Bournemouth University, UK
Zhensheng Guo, Siemens AG, Germany
Ismail Hababeh, German Jordanian University, Jordan
Shahliza Abd Halim, Lecturer in Universiti Teknologi Malaysia, Malaysia
Herman Hartmann, University of Groningen, The Netherlands
Jameleddine Hassine, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Peizhao Hu, NICTA, Australia
Chih-Cheng Hung, Southern Polytechnic State University, USA
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Noraini Ibrahim, Universiti Teknologi Malaysia, Malaysia
Anca Daniela Ionita, University "POLITEHNICA" of Bucharest, Romania
Chris Ireland, Open University, UK
Kyoko Iwasawa, Takushoku University - Tokyo, Japan
Mehrshid Javanbakht, Azad University - Tehran, Iran
Wassim Jaziri, ISIM Sfax, Tunisia
Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Jinyuan Jia, Tongji University. Shanghai, China
Maria Joao Ferreira, Universidade Portucalense, Portugal
Ahmed Kamel, Concordia College, Moorhead, Minnesota, USA
Teemu Kanstrén, VTT Technical Research Centre of Finland, Finland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Ayad ali Keshlaf, Newcastle University, UK
Nhien An Le Khac, University College Dublin, Ireland
Sadegh Kharazmi, RMIT University - Melbourne, Australia
Kyoung-Sook Kim, National Institute of Information and Communications Technology, Japan
Youngjae Kim, Oak Ridge National Laboratory, USA
Cornel Klein, Siemens AG, Germany
Alexander Knapp, University of Augsburg, Germany
Radek Koci, Brno University of Technology, Czech Republic
Christian Kop, University of Klagenfurt, Austria
Michal Krátký, VŠB - Technical University of Ostrava, Czech Republic
Narayanan Kulathuramaiyer, Universiti Malaysia Sarawak, Malaysia
Satoshi Kurihara, Osaka University, Japan
Eugenijus Kurilovas, Vilnius University, Lithuania
Alla Lake, Linfo Systems, LLC, USA
Fritz Laux, Reutlingen University, Germany
Luigi Lavazza, Università dell'Insubria, Italy
Fábio Luiz Leite Júnior, Universidade Estadual da Paraíba, Brazil
Alain Lelu, University of Franche-Comté / LORIA, France
Cynthia Y. Lester, Georgia Perimeter College, USA
Clement Leung, Hong Kong Baptist University, Hong Kong

Weidong Li, University of Connecticut, USA
Corrado Loglisci, University of Bari, Italy
Francesco Longo, University of Calabria, Italy
Sérgio F. Lopes, University of Minho, Portugal
Pericles Loucopoulos, Loughborough University, UK
Alen Lovrencic, University of Zagreb, Croatia
Qifeng Lu, MacroSys, LLC, USA
Xun Luo, Qualcomm Inc., USA
Stephane Maag, Telecom SudParis, France
Ricardo J. Machado, University of Minho, Portugal
Maryam Tayefeh Mahmoudi, Research Institute for ICT, Iran
Nicos Malevris, Athens University of Economics and Business, Greece
Herwig Mannaert, University of Antwerp, Belgium
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Francesco Marcelloni, University of Pisa, Italy
Eda Marchetti, Consiglio Nazionale delle Ricerche (CNR), Italy
Gerasimos Marketos, University of Piraeus, Greece
Abel Marrero, Bombardier Transportation, Germany
Adriana Martin, Universidad Nacional de la Patagonia Austral / Universidad Nacional del Comahue, Argentina
Goran Martinovic, J.J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
Stephan Mäs, Technical University of Dresden, Germany
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Jose Merseguer, Universidad de Zaragoza, Spain
Seyedeh Leili Mirtaheri, Iran University of Science & Technology, Iran
Lars Moench, University of Hagen, Germany
Yasuhiko Morimoto, Hiroshima University, Japan
Antonio Navarro Martín, Universidad Complutense de Madrid, Spain
Filippo Neri, University of Naples, Italy
Muaz A. Niazi, Bahria University, Islamabad, Pakistan
Natalja Nikitina, KTH Royal Institute of Technology, Sweden
Roy Oberhauser, Aalen University, Germany
Pablo Oliveira Antonino, Fraunhofer IESE, Germany
Rocco Oliveto, University of Molise, Italy
Sascha Opletal, Universität Stuttgart, Germany
Flavio Oquendo, European University of Brittany/IRISA-UBS, France
Claus Pahl, Dublin City University, Ireland
Marcos Palacios, University of Oviedo, Spain
Constantin Paleologu, University Politehnica of Bucharest, Romania
Kai Pan, UNC Charlotte, USA
Yiannis Papadopoulos, University of Hull, UK
Andreas Papasalouros, University of the Aegean, Greece
Rodrigo Paredes, Universidad de Talca, Chile
Päivi Parviainen, VTT Technical Research Centre, Finland
João Pascoal Faria, Faculty of Engineering of University of Porto / INESC TEC, Portugal
Fabrizio Pastore, University of Milano - Bicocca, Italy

Kunal Patel, Ingenuity Systems, USA
Óscar Pereira, Instituto de Telecomunicacoes - University of Aveiro, Portugal
Willy Picard, Poznań University of Economics, Poland
Jose R. Pires Manso, University of Beira Interior, Portugal
Sören Pirk, Universität Konstanz, Germany
Meikel Poess, Oracle Corporation, USA
Thomas E. Potok, Oak Ridge National Laboratory, USA
Christian Prehofer, Fraunhofer-Einrichtung für Systeme der Kommunikationstechnik ESK, Germany
Ela Pustułka-Hunt, Bundesamt für Statistik, Neuchâtel, Switzerland
Mengyu Qiao, South Dakota School of Mines and Technology, USA
Kornelije Rabuzin, University of Zagreb, Croatia
J. Javier Rainer Granados, Universidad Politécnica de Madrid, Spain
Muthu Ramachandran, Leeds Metropolitan University, UK
Thurasamy Ramayah, Universiti Sains Malaysia, Malaysia
Prakash Ranganathan, University of North Dakota, USA
José Raúl Romero, University of Córdoba, Spain
Henrique Rebêlo, Federal University of Pernambuco, Brazil
Hassan Reza, UND Aerospace, USA
Elvinia Riccobene, Università degli Studi di Milano, Italy
Daniel Riesco, Universidad Nacional de San Luis, Argentina
Mathieu Roche, LIRMM / CNRS / Univ. Montpellier 2, France
José Rouillard, University of Lille, France
Siegfried Rouvrais, TELECOM Bretagne, France
Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance, Germany
Djamel Sadok, Universidade Federal de Pernambuco, Brazil
Ismael Sanz, Universitat Jaume I, Spain
M. Saravanan, Ericsson India Pvt. Ltd -Tamil Nadu, India
Idrissa Sarr, University of Cheikh Anta Diop, Dakar, Senegal / University of Quebec, Canada
Patrizia Scandurra, University of Bergamo, Italy
Daniel Schall, Vienna University of Technology, Austria
Rainer Schmidt, Munich University of Applied Sciences, Germany
Cristina Seceleanu, Mälardalen University, Sweden
Sebastian Senge, TU Dortmund, Germany
Isabel Seruca, Universidade Portucalense - Porto, Portugal
Kewei Sha, Oklahoma City University, USA
Simeon Simoff, University of Western Sydney, Australia
Jacques Simonin, Institut Telecom / Telecom Bretagne, France
Cosmin Stoica Spahiu, University of Craiova, Romania
George Spanoudakis, City University London, UK
Cristian Stanciu, University Politehnica of Bucharest, Romania
Lena Strömbäck, SMHI, Sweden
Osamu Takaki, Japan Advanced Institute of Science and Technology, Japan
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Wasif Tanveer, University of Engineering & Technology - Lahore, Pakistan
Ergin Tari, Istanbul Technical University, Turkey

Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Jean-Claude Thill, Univ. of North Carolina at Charlotte, USA
Pierre Tiako, Langston University, USA
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Guglielmo Trentin, National Research Council, Italy
Dragos Truscan, Åbo Akademi University, Finland
Chrisa Tsinaraki, Technical University of Crete, Greece
Roland Ukor, FirstLinq Limited, UK
Torsten Ullrich, Fraunhofer Austria Research GmbH, Austria
José Valente de Oliveira, Universidade do Algarve, Portugal
Dieter Van Nuffel, University of Antwerp, Belgium
Shirshu Varma, Indian Institute of Information Technology, Allahabad, India
Konstantina Vassilopoulou, Harokopio University of Athens, Greece
Miroslav Velez, Aries Design Automation, USA
Tanja E. J. Vos, Universidad Politécnica de Valencia, Spain
Krzysztof Walczak, Poznan University of Economics, Poland
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Wojciech Wiza, Poznan University of Economics, Poland
Martin Wojtczyk, Technische Universität München, Germany
Hao Wu, School of Information Science and Engineering, Yunnan University, China
Mudasser F. Wyne, National University, USA
Zhengchuan Xu, Fudan University, P.R.China
Yiping Yao, National University of Defense Technology, Changsha, Hunan, China
Stoyan Yordanov Garbatov, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, INESC-ID, Portugal
Weihai Yu, University of Tromsø, Norway
Wenbing Zhao, Cleveland State University, USA
Hong Zhu, Oxford Brookes University, UK
Qiang Zhu, The University of Michigan - Dearborn, USA

CONTENTS

pages: 205 - 213

Screencasts: Enhancing Coursework Feedback for Game Programming Students Revisited

Robert Law, Glasgow Caledonian University, United Kingdom

pages: 214 - 226

Revision Control and Automatic Documentation for the Development Numerical Models for Scientific Applications

Martin Zinner, Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden, Germany

Karsten Rink, Department of Environmental Informatics, Helmholtz Centre for Environmental Research (UFZ), Germany

René Jäkel, Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden, Germany

Kim Feldhoff, Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden, Germany

Richard Grunzke, Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden, Germany

Thomas Fischer, Department of Environmental Informatics, Helmholtz Centre for Environmental Research (UFZ), Germany

Rui Song, Technical Information Systems, Technische Universität Dresden, Germany

Marc Walther, Department of Environmental Informatics, Helmholtz Centre for Environmental Research (UFZ), Germany

Thomas Jejkal, Institute for Data Processing and Electronics, Karlsruhe Institute of Technology (KIT), Germany

Olaf Kolditz, Department of Environmental Informatics, Helmholtz Centre for Environmental Research (UFZ), Germany

Wolfgang E. Nagel, Center for Information Services and High Performance Computing (ZIH), Technische Universität Dresden, Germany

pages: 227 - 238

Versatile but Precise Semantics for Logic-Labelled Finite State Machines

Callum McColl, Griffith University, Australia

Vladimir Estivill-Castro, Griffith University, Australia

Rene Hexel, Griffith University, Australia

pages: 239 - 246

Pragmatic Approach to Automated Testing of Mobile Applications with Non-Native Graphic User Interface

Maxim Mozgovoy, University of Aizu, Japan

Evgeny Pyshkin, University of Aizu, Japan

pages: 247 - 275

Deriving Learning Strategies from Words Lists: Digital Dictionaries, Lexicons, Directed Graphs and the Symbol Grounding Problem

Jean-Marie Poulin, Université du Québec à Montréal, Canada

Alexandre Blondin Massé, Université du Québec à Montréal, Canada

pages: 276 - 285

Subjective Assessment of Text Quality on Smartphone Display with Super Resolution

Aya Kubota, Kogakuin University, Japan

Seiichi Gohshi, Kogakuin University, Japan

pages: 286 - 298

Multi-dimensional Context Creation Based on the Methodology of Knowledge Mapping

Claus-Peter Rückemann, Westfälische Wilhelms-Universität Münster (WWU); Knowledge in Motion, DIMF; Leibniz Universität Hannover, Germany

pages: 299 - 310

An International Survey of Practitioners' Views on Personas: Benefits, Resource Demands and Pitfalls

Engie Bashir, Middlesex University Dubai, United Arab Emirates

Simon Attfield, Middlesex University, United Kingdom

pages: 311 - 322

Context-aware Storage and Retrieval of Digital Content: Database Model and Schema Considerations for Content Persistence

Hans-Werner Sehring, Namics, Germany

pages: 323 - 334

A Novel Training Algorithm based on Limited-Memory quasi-Newton Method with Nesterov's Accelerated Gradient in Neural Networks and its Application to Highly-Nonlinear Modeling of Microwave Circuit

Shahrzad Mahboubi, Graduate school of Electrical and Information Engineering, Shonan Institute of Technology, Japan

Hiroshi Ninomiya, Graduate school of Electrical and Information Engineering, Shonan Institute of Technology, Japan

pages: 335 - 346

Analyzing Collaborative Learning Process by Deep Learning Methods: A Multi-Dimensional Coding Scheme with an Assessment Model

Taketoshi Inaba, Tokyo University of Technology, Japan

Chihiro Shibata, Tokyo University of Technology, Japan

Kimihiko Ando, Tokyo University of Technology, Japan

pages: 347 - 357

Visibility Velocity Obstacles (VVO): Visibility-Based Path Planning in 3D Environments

Oren Gal, Technion, Israel

Yerach Doytsher, Technion, Israel

pages: 358 - 367

A Comprehensive Workplace Environment based on a Deep Learning Architecture for Cognitive Systems

Thorsten Gressling, ARS Computer und Consulting GmbH, Germany

Veronika Thurner, Munich University of Applied Sciences, Germany

pages: 368 - 378

Novel Field Oriented Patient Monitoring Platform for Healthcare Facilities

Yoshitoshi Murata, Iwate Prefectural University, Japan

Rintaro Takahashi, Iwate Prefectural University, Japan

Tomoki Yamato, Iwate Prefectural University, Japan

Shohei Yoshida, Ilwate Prefectural University Graduate School, Japan
Masahiko Okamura, Ilwate Prefectural University Graduate School, Japan

pages: 379 - 389

Compositing “Stand Off” Ground Penetrating Radar Scans of Differing Frequencies

Roger Tilley, University of California, Santa Cruz, United States of America
Hamid Sadjadpour, University of California, Santa Cruz, United States of America
Farid Dowla, University of California, Santa Cruz, United States of America

pages: 390 - 399

Earth Observation Semantics and Data Analytics for Coastal Environmental Areas

Corneliu Octavian Dumitru, DLR, Germany
Gottfried Schwarz, DLR, Germany
Mihai Datcu, DLR, Germany

pages: 400 - 417

Automated Continuous Data Quality Measurement with Qualle

Lisa Ehrlinger, Johannes Kepler University Linz and Software Competence Center Hagenberg GmbH, Austria
Bernhard Werth, Johannes Kepler University Linz, Austria
Wolfram Wöß, Johannes Kepler University Linz, Austria

pages: 418 - 439

Protecting Against Reflected Cross-Site Scripting Attacks

Pål Ellingsen, Western Norway University of Applied Sciences, Norway
Andreas Svardal Vikne, Western Norway University of Applied Sciences, Norway

pages: 440 - 451

Dynamic Programming Approach to Retrieving Similar Candlestick Charts for Short-Term Stock Price Prediction

Yoshihisa Udagawa, Tokyo Polytechnic University, Japan

pages: 452 - 465

POMVCC: Partial Order Multi Version Concurrency Control

Yuya Isoda, Hitachi, Ltd., Japan
Atsushi Tomoda, Hitachi Ltd., Japan
Tsuyoshi Tanaka, Hitachi Ltd., Japan
Kazuhiko Mogi, Hitachi Ltd., Japan

Screencasts: Enhancing Coursework Feedback for Game Programming Students Revisited

Robert Law

School of Computing, Engineering and Built Environment
Glasgow Caledonian University
Glasgow, Scotland
Email: robert.law@gcu.ac.uk

Abstract—Feedback is an important part of learning and, as such is vital for students to develop and progress throughout their academic life. Programming can be an abstract concept that students find challenging to comprehend therefore good feedback is important to their progress and their motivation to continue programming. This paper will discuss the process of enhancing coursework feedback for Game Programming students through the use of screencasts. The hypothesis being that game programming by its nature is audio-visual thus, providing feedback using an audio-visual medium should increase the student's perception of their feedback such that it is perceived to be clearer, easier to comprehend and personalised.

Keywords—Screencasts; Feedback; Software Development.

I. INTRODUCTION

Following on from work done by Law [1]: this paper revisits the concept of enhancing coursework feedback for Game Programming students through the use of screencasts with a view to offering a template that can be utilised in the production of screencasts, which both minimise the Lecturer's work load and maximises the students feedback.

The United Kingdom's (UK) National Student Survey (NSS) [2] is a survey for final year students at all of the UK's publicly funded Higher Education Institutions (HEIs) and is administered by Ipsos MORI. The NSS comprises of 27 questions across eight categories attempting to capture the students learning experience. The NSS acts as a barometer of student satisfaction and thus, is an influential survey giving the student body a collective voice. The data from the survey is publicly available and is used by prospective students when choosing their potential University.

This survey has a number of different sections, one of, which is Assessment and Feedback. The perennial view from students suggests that there is scope for improvement with regard to Feedback. Comparing all eight categories it can be seen that Assessment and Feedback is continually at the bottom. This would suggest that there is still room for improvement. Table I shows all the sections of the questionnaire and their corresponding percentage satisfaction rating. It is noticeable, from Table I, that satisfaction with Assessment and Feedback is between 5 and 14 percentage points behind 7 of the 8 remaining categories suggesting that the students' impression of feedback and the instrument of feedback delivery have not met entirely with the students' expectations [3], [4].

Viewing the statistics on a nation by nation basis against the UK average creates an interesting picture of how students

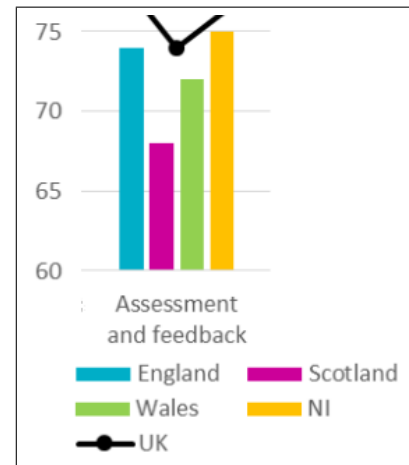


Figure 1. Assessment and Feedback results 2016 by nation

in each of the four nations differ in their perceptions of the level of feedback they receive. Figure 1 shows a comparison of all four nations. Working in an academic institution in Scotland the picture painted is somewhat alarming with Scotland six points below the UK average [5]. The Assessment and Feedback section of the survey is comprised of five questions; two relating to assessment and three relating to feedback. The feedback questions are shown in Table II. The questions in Table II emphasize the students' desire for expeditious, clear and detailed feedback [6].

TABLE I. PERCENTAGE SATISFACTION ACROSS CATEGORIES FROM NSS QUESTIONNAIRE

Categories	2015	2016
The teaching on my course	87	87
Assessment and feedback	73	74
Academic support	82	82
Organisation and management	79	79
Learning resources	85	86
Personal development	83	82
Overall satisfaction	86	86

The remainder of this paper is organized as follows: Section II will provide an overview of the author's rationale for the use of screencasts within the feedback process; indicating the nature of the cohort and the subject area studied. Section III will provide information about pedagogical issues related to screencasting, Section IV offers an introduction to the

TABLE II. EXTRACT OF FEEDBACK QUESTIONS FROM NSS QUESTIONNAIRE

<i>Feedback Questions asked as part of NSS</i>
Feedback on my work has been prompt.
I have received detailed comments on my work.
Feedback on my work has helped me clarify things I did not understand.

technologies available for screencasting. Section V presents an overview of the screencasting process, while Section VI reflects on the informally gathered feedback from the student cohort. Section VII discusses issues encountered by the author during the creation of the screencasts. Section VIII attempts to derive conclusions based on the synthesis of Sections III, IV, VI and VII. Section IX offers ideas for future work.

II. RATIONALE

Teaching programming, and in particular, game programming it can be difficult to offer students feedback on coursework submissions that are not either too generic and brief or ultimately too verbose and overcomplicated. Getting the balance of written feedback correct can be a daunting task. Thompson and Lee [7] suggest that feedback is “a pedagogical tool to improve learning by motivating students to rethink and rework their ideas rather than simply proofread and edit for errors.” Interestingly, Thompson and Lee [7] quote Notar, Wilson and Ross that “feedback should focus on improving the skills needed for the construction of end products more than on the end products themselves”. This particular observation is very apt for teaching programming concepts and programming languages as the feedback given is in the context of the students programming skills rather than their end product, in this case their game. The feedback is intended to improve the students ability to produce structured, economical code and illustrate the necessary skills for debugging program code.

The author teaches game programming modules at various levels within the undergraduate programme BSc (Honours) Game Software Development. It would seem natural for game programming students who primarily work with a very audio visual medium to receive feedback for their programming coursework as an audio-visual screencast. It was therefore decided to implement a trial with a second year cohort undertaking the module Game Programming 1. This module was chosen as it was a core module for both the Game Software Development students and the Game Design students. The module introduces students to coding using C++ and OpenGL with the emphasise on the production of a 2D game prototype. The module had approximately 70 students participating in it with a near even split of Game Software Development and Game Design students.

The coursework required the students to create a game of their choosing. The coursework specification provided the students with a number of requirements that had to be met and a marking scheme was provided as a guide to the aesthetic appearance of their game and the functional aspects of the underlying code.

This paper will focus on post coursework feedback, which, in this case represents feedback given to the student after completion of the module. The submission date for the coursework is normally the last week of term, therefore, feedback would normally be provided in a written format, distributed via email.

The aim of the research is twofold: to better understand the delivery of feedback to students undertaking programming courses via the medium of screencasts such that the students feel that they have gained a meaningful commentary on their coursework submission, which will, hopefully, lead them to improve their subsequent submissions; and to identify a process or template that can be used by Lecturer's to minimise their work load while maximising the amount of feedback given to the student.

III. PEDAGOGICAL ISSUES

So what is a screencast? For the purposes of this paper a screencast will be defined as a recording of the current content of the computer screen with an audio narration providing relevant commentary, i.e., feedback [8], [9], [10]. For this reason Atfield-cutts [11] suggests that “... video feedback potentially, such a powerful enabler for programming students in particular.” As part of their learning it is important for students to receive feedback on any of the work that they produce. However, Atfield-cutts [11] identifies that “Student engagement with feedback is often lacking and in that case, a valuable learning opportunity is missed.”, thus, it is important to find ways in, which, students can be encouraged to be part of the feedback loop.

It is postulated by Thompson and Lee [7] that student reluctance to engage with the feedback process maybe due to an attempt to create an equilibrium between study, home and work life, hence, Atfield-cutts [11] suggests that, in order to re-engage the students with the feedback process, the process itself must be perceived by the students to require less time and /or effort, or it must be deemed more pleasant and/or useful by students.

Race [12] identifies a number of common formats used to disseminate feedback to students: handwritten, word processed, model answers/solutions, rubric proformas, oral feedback, email and computer marked assessment. These methods can be issued individually or as general feedback based on the performance of a cohort or group.

Race [12] suggests five attributes of feedback: Timely, intimate and individual, empowering, open doors not close them and manageable. Timely feedback is a goal that is highly desired and greatly prized, but, can be dictated by class size or other commitments. Intimate and individual feedback should reflect the student's own submission. Empowering feedback is harder to achieve, as it is a balancing act between positive feedback and a critic, warts and all, of the student's submission. Open doors, not close them refers to the use of language within feedback and the expectation this can set for the student and the feedback they receive for their next submission. Manageable, is viewed from the perspective of both the student and the lecturer, i.e., the effort expended by the lecturer to produce the feedback and the volume of feedback received by the student could cause them to miss something important [12].

Using the written word to provide annotated feedback to students can be taken out of context [10] and therefore the benefit of the feedback can be lost. Worse still, the feedback taken out of context can be misconstrued as a criticism of their work [9] rather than a pointer to improvement. The loss of visual and aural cues, which aid understanding [13], from the written feedback process is therefore something that

screencasting can help combat. Moreover, screencast feedback has the potential to "provide more information to students about their work compared to the written commentary." [14]

As part of Evans [15] "12 pragmatic actions" for effective feedback, one suggestion is for students to be presented with an early assessment opportunity such that they can receive early feedback, which, can be built upon prior to final submission.

It has been mooted that audio-visual screencasts can create for the Lecturer the concept of "social presence" and "an opportunity for conveying positive encouragement through intonation." [9]. This ability to use intonation to emphasize important [3] aspects of feedback make the use of screencasts a benefit for the student. Indeed, Seror [16], believes that screencast feedback offers the ability to provide "a more conversational and personal form of feedback." Couple this with the ability to hear the feedback in the manner the Lecturer intended it and the loss of the visual and aural cue associated with face to face feedback are somewhat restored. The volume of information that can be presented to the student via the audio aspect of screencasts is far larger than written feedback alone and in a shorter time period [3], [17], [18].

Galanos et al. cite the use of screencasts as a method of giving a student personalised feedback by recording the lecturer debugging the students program code while commenting on it [19]. Also suggested is the use of an attached webcam to offer "picture in picture" of the lecturer while debugging the program code, helping to offer that personal touch [19].

It has been suggested that screencasts can aid the student's understanding of their feedback by negating the need for continual cross-referencing between feedback and assessment and secondly the use of conversation style feedback rather than a more formal written academic feedback [9]. It has also been suggested that students find it clearer to "understand the marker's reasoning" [8] and comments [20] when presented in a screencast.

Clarity of feedback is important to students [21]; they do not want to receive feedback that could be deemed "vague, unclear and confusing" [22]. Thus, the audio-visual nature of screencasts can help enrich the feedback pinpointing unambiguously exactly what is being commented on [22]. The promptness or timeliness of feedback is another concern for students as evidenced by the low scores in the National Student Survey [5]. Hope suggests that educators are under an "obligation to provide meaningful feedback within a reasonable timeframe" [3]. Mathisen proffers anecdotal evidence from the field that screencasts can provide more feedback and can be produced in less time [22].

It has been mooted by O'Malley that one of a quartet of criteria needed for feedback to be effective is for it to be personal [23]. Screencasting offers the student personalised feedback that is tailored to their submission [9]. Chewar and Matthews state that the use of screencasts to provide feedback allows for more detailed, accurate and robust feedback [24]. Thomson and Lee also suggest that feedback given through the use of screencasts has the capacity to motivate and boost the students engagement with their learning [7].

Sugar et al. [25] undertook to research the anatomy of a screencast with a view to developing a framework for the production of screencasts to better aid Lecturers in producing effective learning screencasts. Although this research was

focused on the production of screencasts for learning e.g., how to save a spreadsheet as a CSV file, this framework offers potential for the production of feedback screencasts.

Sugar et al. [25] framework consists of two categories: Structural elements and instructional strategies. These categories are further subdivided as follows: Structural elements comprises of "bumpers, screen movement and narration"; Instructional elements comprises of "provide overview, describe procedure, present concept, elaborate content, and focus attention." Figure 2 shows the framework with a further layer of subdivision.

Examining each of the structural and instructional elements suggests that this framework could be adapted to reflect the creation of feedback screencasts. Although, the intention of feedback screencasts is to add a level of personalisation, a framework or checklist would act as a valuable guide to the desired content of a feedback screencast.

The three structural components offer a clear set of tools for adding a degree of personalisation to a feedback screencast. For example, bumpers, a term borrowed from radio broadcasts [25], is a technique used to offer a salutation and/or a valediction to the screencast. This allows the Lecturer to provide an opening and closing greeting to the student e.g., possible opening statement

Hi Jim, Well done on completing the coursework! I will now provide you with feedback on your submission, which will hopefully prove useful.

and a possible closing statement

Jim, I have covered a number of aspects in your submission and I hope that the feedback has helped elaborate on the key aspects of the coursework and how your submission met that criteria. Thanks, Bobby.

The examples above exemplify the type of personalisation that can be applied to the feedback screencast.

Screen movement can be split into two types: static or dynamic; static screen movement is "a constant frame in, which the cursor moves within that frame" and dynamic screen movement is "the capture frame moves around the screen, keeping the cursor in the center." [25] A mixture of both types could be used for various aspects of programming feedback, for instance, while playing the student's game static screen movement would be appropriate, but, providing feedback on the student's code would benefit from the use of dynamic screen movement allowing the Lecturer to hone in on the desired code fragment.

Narration is an important aspect of a feedback screencast as it is the Lecturer's route to personalisation. Sugar et al. [25] define narration as explicit and implicit; explicit narration depicts what can be seen on screen and implicit narration refers to more generalised commentary. A combination of both would be appropriate for a feedback screencast.

The five instructional components offer a set of tools, which can be mixed and matched were appropriate to add the necessary degree of personalisation to a feedback screencast. As noted by Sugar et al. [25], not all of these instructional components were found in instructional screencasts, therefore, not all of these components will be required in a feedback screencast.

In the context of instructional screencasts "Provide Overview" delivers the "necessary background information that learners need in order to understand the context and/or the purpose of the screencasting topic;" [25]. This approach is also feasible for feedback screencasts as it seems appropriate to indicate to the student what the assessment was designed to test. Seror [16] exemplifies this approach saying "I typically start a recording with a few brief words about what I will be focusing on and how the feedback will proceed ..."

When feeding back to a student about the particular way they have coded their game "describe procedure" allows the Lecturer the ability to take an aspect of the coursework and relate the appropriate programming technique required to satisfactorily implement it.

The Lecturer can use the "present concept" strategy to identify sections of the student's code explaining how their code could have been improved or optimised through rearranging the code segment, aligning it to a design pattern, and explaining why it is a better solution.

The concept behind "focus attention" is to use a combination of the mouse pointer and narration to draw the student's attention to an area of their code that has been implemented to a high standard or could be improved. This could also be enhanced by using dynamic screen movement.

The final instructional component "elaborate content" is an opportunity for the Lecturer to "enrich" the student's comprehension and provide the student with alternative approaches that will expand their learning [25].

Subsequently, having examined Sugar et al. [25] screencast framework for instructional screencasts, it is evident that this framework is a viable framework for use with Feedback screencasts.

IV. TECHNOLOGY

There are a number of different combinations of hardware and software that can be used to create a screencast. The following sections will describe the hardware and software used by the author to create feedback screencasts.

A. Hardware

To capture good quality audio it is advisable to refrain from using the built-in device microphone but instead opt for a headset or external microphone [3], [26], [27]. The benefit of using a headset is the consistent distance from the mouth [28] and the ability to position it slightly below the mouth to minimize the noise of breathing [26].

B. Software

A number of software packages are available and these range from desktop applications to web based applications, which, in turn, vary in price from free to hundreds of pounds [28].

Table III illustrates a small selection of available screen-casting software including a brief description of the software, highlighting its main features, has been provided along with a web link to the software. Kilickaya [14] cautions for the need to select screencast software wisely, suggesting "A benefit-cost analysis should be conducted before making the choice." It is worth noting that free software may well suffer from

limitations of functionality or may not have some of the desirable advanced features of their paid for counterparts [14].

Software used for this paper was Screencast-O-Matic a web based application offering a limited version free. The free version allows up to 15 minutes of recording, recording from screen and webcam, the ability to publish to YouTube and the ability to save in popular formats such as .MP4, .AVI and .FLV. It is relatively easy to use [10] and has a very handy countdown before recording begins.

V. RECORDING SCREENCAST FEEDBACK

Although the screencast in this instance is being created in response to an unknown entity it is still important to apply the rules of creating instructional screencasts by planning [28]. Planning is very important [29] as there will be a number of areas that will require feedback. For the game produced by the students the coursework feedback was broken into the following areas: aesthetics, game play, code structure and compilation. Each of these areas was broken down further with key points: aesthetics covered the games look and feel and interface design; game play covered the ease and enjoyableness of the game, responsiveness of game objects to keyboard/gamepad interaction; code structure covered neatness, use of the fundamental programming building blocks, use of language features, data structures, and the object oriented paradigm; compilation covered the programming compiling and the appropriate use of compilation switches.

Unlike recording a conventional educational screencast there is no need to produce a script [30] as the coursework submissions will not be predictable and a script can depersonalize the feedback and make it feel unnatural [4]. Armed with the marking scheme and the aforementioned plan the process of creating the screencast could be started. A number of considerations were taken into account before commencing the screencast process:

- Determining a location, which has a low level of background noise [26] and little chance of being interrupted.
- Use a good quality headset, positioning the microphone slightly below the mouth [26].
- Switch off any software that activates pop ups such as email, Facebook or instant messenger as these could end up being recorded [4].
- Use and stick to the devised plan for consistency.
- Speak naturally and positively [30] making good use of intonation [3].
- Use of the pause button [28] at the end of each section to allow time to gather one's thoughts prior to the start of the next section.
- Screencast duration should be between five and ten minutes [14].

Having evaluated Sugar et al. [25] screencast framework it seemed like a logical decision to incorporate the framework into the production of the feedback screencasts.

"Bumpers" were incorporated, allowing a quick introduction to the student, using their name, further using the approach of "Provide Overview" aspect of the framework, explaining the key aspects of the marking scheme being used. At the

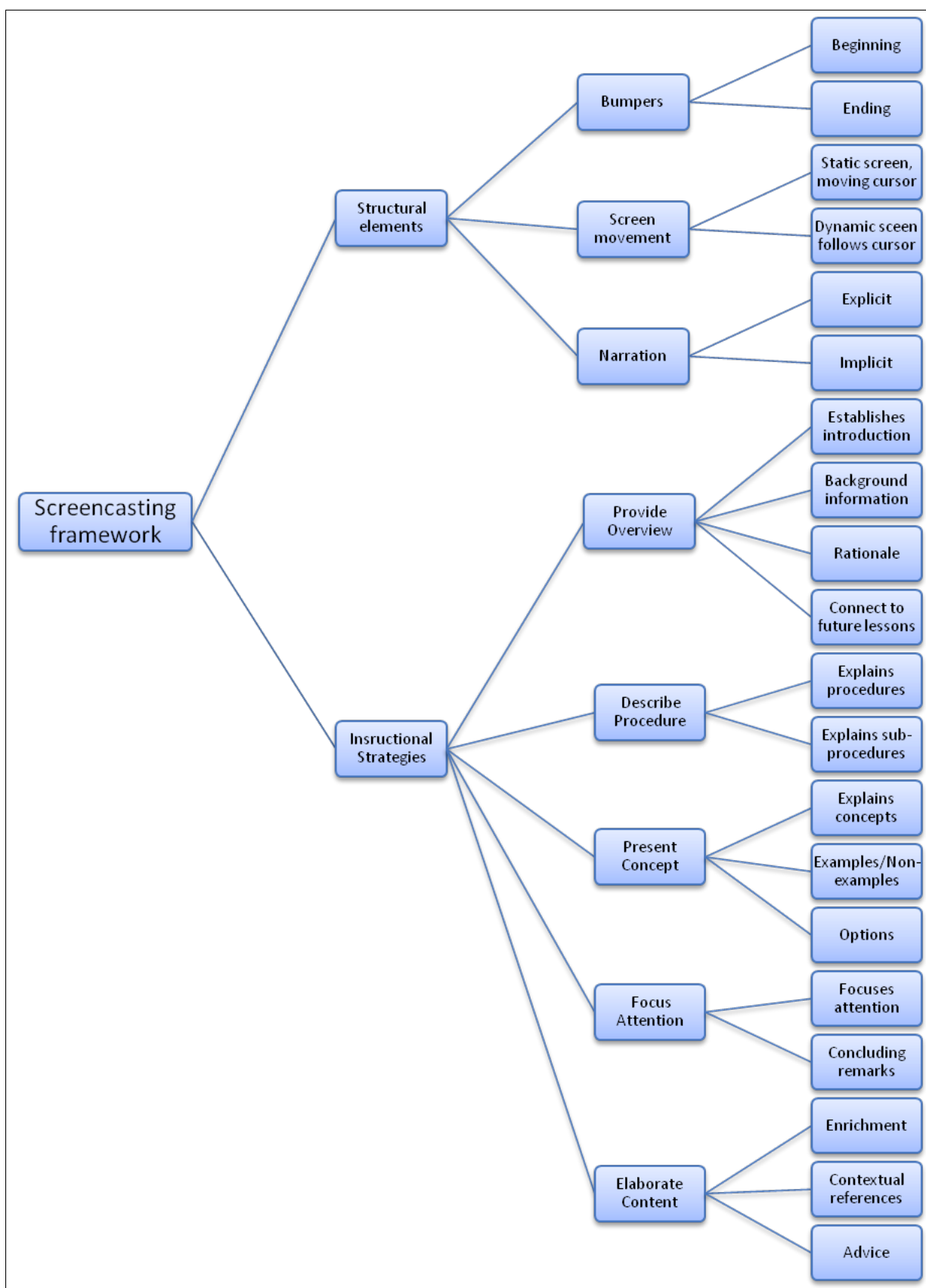


Figure 2. Sugar et al. [25] Screencast Framework

TABLE III. SELECTED SCREENCASTING SOFTWARE

Software	Description	URL
ScreenCast-O-Matic	Three plans: Free (Features Limited to 9 from 18), Deluxe and Premier (paid for). Hosting and Sharing available across all plans, but, with varied restrictions. Editing Tools available for Deluxe and Premier plans.	https://screencast-o-matic.com/
Snagit	Commercial Software; Requires payment (Free Trial available); Discount for Education; Screen capture, recording and in built editor	https://www.techsmith.com/screen-capture.html
Camtasia	Commercial Software; Requires payment (Free Trial available); Discount for Education; Screen capture, Recording, video editor; can include interactive quizzes	https://www.techsmith.com/video-editor.html
Jing	Freeware; time limited and basic	https://www.techsmith.com/jing-tool.html
Screencastify	Chrome extension; Saves to Google Drive, YouTube, Export as .MP4: Mouse focus, Draw with Pen, Embed Webcam; Free and paid for version.	https://www.screencastify.com/

end of the screencast, the student, again using their name, is given a summary, the key criteria of what was being assessed, their coursework performance and encouraged to contact the Lecturer if they would like anything explained further, which again, illustrates the concept of the "Provide Overview" aspect of the framework.

"Screen Movement" within the feedback screencasts was mainly static, but, an attempt was made to use dynamic screen movement to focus in more tightly on particular code segments and when showing the debugging process. Static screen movement is applicable when recording the students game running as the whole screen is visible. During the recording process all mouse movements and clicks are visible to the viewer as a large coloured circle that will change colour when the mouse button is clicked. This is exceptionally useful for giving the student unambiguous and precise feedback on their user interface design and layout pointing out what is considered good and what needs improving. Seror [16] outlines virtually this approach as part of a "work flow" adhered to when producing a feedback screencast; my typical work flow begins with opening the file that contains a students assignment record the full screen of my computer. Using a headset microphone, I then begin to read and comment on the text orally and visually. All oral comments are recorded in synch with my mouse movements as I highlight and/or edit various sections of text.

"Narration" is a key aspect to recording a useful and engaging feedback screencast and it is important to synchronise the narration with any mouse actions as this will help "Focus attention" of the student to any aspect of feedback and praise being delivered. As far as possible, every attempt was made to keep the narration explicit such that the student was left with no ambiguity about the comments being made. A tip well worth remembering when narrating a screencast is, to use the pause button on a regular basis, as this will allow time to survey and reflect on the next aspect of the marking scheme before proffering any feedback [16]. Also, as far as possible refrain from the use of implicit descriptions as this can lead to a level of ambiguity that will not be beneficial to the student. Again, during the narration it is useful to utilise the "Provide Overview" aspect of the framework to highlight to the student what is being assessed and how well they met this criteria.

The following structural aspects of the framework are not necessarily required for all students, but, are very useful for weaker students who have submitted a coursework, which has not met the desired criteria for a pass or is a borderline pass. These aspects are best used in combination to delivery a more meaningful feedback experience.

The first of these structural framework aspects is "Describe

procedure"; this can be used to detail a number of different elements of coursework; from how to implement a simple state machine using a case statement to the required steps within the IDE to debug the student's code correctly and effectively. Using the examples presented in the previous sentence showing the implementation of a state machine would require the student's code to be rewritten with a full explanation of why their code is incorrect and how the changes made to their code implement the state machine correctly. Likewise, for student submissions that did not execute, a debug process could be illustrated showing the debug process and a suitable narration, which, along with the required mouse clicks to access the appropriate menu options in the IDE, that would hopefully allow the student to solve a similar problem if encountered again. This ability to show a debug process in operation is a valuable process that merits a role out to all students as the ability to debug code is a valuable skill.

The second structural framework aspect is "Present concept"; this can be used to explain how collision detection works with regard to two sprites colliding. Again, examining the student's code and making the requisite changes while offering a suitable explanation of both the concept of collision detection and the code needed to implement it.

The third structural framework aspect is "Elaborate content"; Sugar et al. [25] suggest that this is the point at, which the screencast can "enrich learners' understanding and to encourage learners to consider other aspects of the process or concept". This aspect can be used to illustrate to the student how the concepts and techniques used to master the coursework can be embellished and reused in future courseworks and beyond.

The neatness and compactness of the actual code itself is an important aspect of any programming thus, the screencast gave the author the ability to highlight selected code within the Integrated Development Environment (IDE), in this case Microsoft Visual Studio, offering an audio narrative explaining clearly any deficient code and a visualisation of how the code could be reworked in order to make it neater and more efficient. Good examples of student work could also be highlighted and the student commended for its use.

Most of the screencasts were between 5 and 10 minutes in length depending on the game produced and the exhibited programming ability of the student, which is in keeping with the surveyed literature. Unlike the previous incarnation of the first trial [1] where he feedback screencasts were compressed into a .zip file and returned to the student via e-mail the decision was made to use the author's virtual learning environment (VLE), a version of Blackboard, to upload the video files directly to the student's secure storage area.

VI. STUDENT FEEDBACK

Having applied Sugar et al. [25] Screencast framework to the production of the screencasts the hope was that the student cohort would engage with the feedback screencasts, appreciating the audio-visual nature and the structured approach. Therefore it was heartening that the initial feedback from the students was, on the whole, positive and illuminating with regard to refining the screencast feedback process.

Although, comments were elicited from all (70) students in an informal manner, students were asked to complete a short Google Forms questionnaire, which, comprised of likert style questions and a short response question allowing them to give their initial impression of receiving feedback in this manner. The questions in the questionnaire were based on questions created by Ali [31]. Table IV shows the questions and the percentage, greater than or equal to 3 on the likert scale, responses. Of the 70 students who undertook the module there was 60 respondents to the questionnaire. As all students received both written feedback and feedback in the form of screencasts this allowed the students to compare and contrast the two forms of feedback proffering their thoughts.

Again, as with Law [1] the respondents overwhelming feeling was a sense of personalization and tailoring of feedback to their needs. Students were also receptive to the visual code analysis they received indicating that they understood more readily the need for well written, neat and compact code. Although, anecdotal, the quotes from students help to articulate their view of screencasts for feedback.

"The combination of seeing where I went wrong with Bobby's audio was very useful."

"More personal feedback with clear direction on where I went wrong."

"Seeing my game played by Bobby and with his comments really brought home to me where my interface was lacking."

As with Law [1], the positive feedback suggests that the technique is worth persevering with and a further attempt will be made to hone the screencast framework prior to rolling out screencasting as a delivery mechanism for feedback. The Google Forms questionnaire will be restructured to aid with the capture of qualitative data.

VII. ISSUES

From the perspective of the lecturer there are some issues that need to be addressed. Seror [16], identifies that incorporating screencasts into his teaching "required adjustments to my regular feedback practices."

Firstly, the time taken to prepare the screencast feedback does not necessarily equate to the actual time of the screencast that the student will observe, which tallies with Mathieson [13], whom identifies this as an "important caveat" backing this up by noting that, during her trials, screencasts took approximately twice the time to produce. This is not, necessarily, due to the screencast being edited but the time taken to record the screencast itself. Kilickaya [14] suggests that the time taken to record screencast feedback may well be dependant on "the type of written work being marked as well as the comprehensibility of feedback."

Although, in Section V, a key piece of advice is to plan and prepare for the screencast by using some form of rubric, the application of this rubric can leave the recording having a staccato and unnatural feel. A solution to this is to pause the recording after each section and compose oneself before recording the next section. This will add time to the process but will prove worthwhile in the long term. The expectation would be, as noted by Atfield-cutts [11], that "the process sped up with practise to the point."

Secondly, the time consumed by planning, stopping short of scripting, the feedback. Implementing the screencast framework is not a quick process as this framework needs to be mapped against the coursework/assignment noting the key points of learning that should be fed back to the student, if and when appropriate. Again, this can be countered by the regular use of the technique and the fact that there may well be overlap between assignments, which, will lend itself to the re-use of some key points.

Thirdly, choosing a suitable location to record the screencasts is imperative as interruptions not only break the lecturers concentration but also can be inadvertently recorded thus, requiring the recording to be edited or, worse still, to be scrapped. A quiet location devoid of interruptions is not always possible in a busy University. It is not an insurmountable challenge but definitely something to be aware of prior to starting any recordings.

A fourth issue is the size of the recorded screencasts with regard to the required disk storage. The size is dependant on a number of factors including: video codec used, screen size being recorded, and resolution of recording. For example a screencast recorded using the H.264 video codec for YouTube with a definition of 720p, a resolution of 1280x720, 25 frames per second and lasting 5 minutes will require approximately 1.73 gigabytes of disk space. Thus, for a cohort of 70 students, approximately 121 gigabytes of disk storage would be required. This leads to a secondary issue with the delivery mechanism used for distributing the recordings to the students. Distribution by email can be a problem as there may be a restriction on the maximum file size that can be attached to an outgoing email. If this is the case then an alternative method will be required; this could be by uploading the file to a Managed Learning Environment (MLE). Cognisance should also be taken with regard to the time taken to return the feedback to the students [14] as it is not a trivial task to return sizeable video files.

All of the aforementioned issues are solvable with a bit of careful planning and preparation prior to embarking on the recording process.

VIII. CONCLUSION

Results from this second run of the project suggest that screencasts are, tentatively, potentially of benefit to students, but, may incur a time overhead for staff. From a student point of view, this would go along way to addressing the students perception of feedback as highlighted by the UK's National Student Survey.

Reflecting on the creation of the feedback screencasts, it is an interesting exercise to return to the five attributes of feedback, as defined by Race [12], and attempt to analyse, albeit subjectively, if screencast feedback can be thought of

TABLE IV. SURVEY RESULTS

Question	≥ 3
Did you feel receiving feedback through screencast videos helped you understand the programming techniques you implemented?	90%
Did you feel receiving feedback through screencast videos helped you improve your use of the C++ Standard Template Library?	76%
I found screencast videos helpful because I can replay the video at any time.	97%
I found screencast videos helpful because I can pause the video and reflect on how the code could be improved.	92%
I found screencast videos helpful as I understand where I have lost marks.	84%
The audio of the lecturer in the videos was clear.	95%
The language used giving the feedback was easy to understand.	93%
The Lecturer praised the positive aspects of my code.	87%
The feedback was supported by suggestions for improvement of my code.	88%
Watching screencast videos is time-consuming.	28%
I had difficulty loading the videos.	7%
I felt that receiving feedback through screencast videos engaged me actively in the process of code review and optimisation.	78%
I have a positive attitude toward receiving feedback through screencast videos	92%
Did you feel the feedback using screencast videos added a personal aspect?	94%

as improving these attributes. Again, this can be a time consuming exercise.

Timely feedback can be considered as a property of the turnaround time from student submission of coursework to the lecturer returning feedback to the student; to this end screencasting has no influence on this attribute.

Intimate and individual feedback is an interesting attribute; screencasts can help to achieve this attribute, especially for programming, as the student will receive feedback on their programming code, hearing and seeing the lecturer discuss various aspects of their game's code. Empowering feedback is a balance between providing positive feedback and being able to critic the student's work in such a manner that they feel engaged and enthused to progress and push forward. Screencasting feedback can provide the student with the necessary aural and visual cues to afford them the understanding of what is good with their work but also, in a positive manner, how their work can be improved. This is especially good for programming as it is important for students to understand that code that works can still be improved to make it more efficient and that this is a learning process and not a criticism. Open doors, not close them is a delicate area but with a judicious use of appropriate language and the correct vocal intonation the student can be presented with aural cues and, to a certain extent, visual cues that will allow them to synthesise the intended tone of the feedback.

Finally, Manageable, as noted by Race [12] has two aspects: the level of work involved for the lecturer and the volume of feedback given to the student. With regard to the level of work involved for the lecturer this may fluctuate depending on the cohort and the quality of their submissions, therefore, it is possible that it could add somewhat to the lecturers overhead for producing feedback. Hope for faculty would be that the process of creating the feedback screencasts would speed up with each iteration. However, for students, they should have a targeted and enhanced quality of feedback, which should not overburden them but provide the important aspects of the desired feedback they need to progress and improve.

Screencasts provide resource-rich feedback for students combining both narration and visual aspects to enrich and augment traditional feedback practices [16]. The increased feedback that can be crammed into a 5 minute screencast is more personal, clearer, less ambiguous than traditional written feedback and offers to show students "how to fix their own code or use a better technique, directly, without having to direct

them to a generic example." [11], which would seem like a boon for the student. Although, oral feedback is given during lab sessions, this type of feedback is relevant in situ but, when the student refers back to this type of feedback it is entirely at the mercy of the student's ability to accurately record it. In contrast, the student can play and replay the video as many times as they like and the feedback will always be viewed as it was intended. The time to produce the screencasts varies by student submission but on the whole it was surprisingly quick in comparison to written feedback of the same depth.

IX. FUTURE WORK

The intention is to repeat the screencast feedback in the next academic year. The number of students undertaking the module will, again, be in the region of 60 students and should offer a suitable number for judging the timeliness of producing feedback screencasts. The hypothesis is that the experience from this first large scale implementation will lead to a more effective and quicker production process for each screencast and the students will benefit from clear, concise and helpful feedback.

The feedback screencasts will additionally be augmented by including webcam footage of the Lecturer, this will add back the visual and body language cues gained from face to face feedback [13], in the belief that it will "maximise the potential benefits of video feedback" [11].

The module is 12 weeks in duration and students will be asked to submit work at the end of week 8 and also at the end of week 12. Screencast feedback on their week 8 submission will be returned by week 10, which, should allow for the students to benefit from the feedback prior to their final submission in week 12 [15]. After receiving the feedback screencasts the students will be surveyed to ascertain a better representation of their feeling towards this feedback mechanism. Screencast feedback will be returned approximately 10 working days after week 12 submission and should serve to inform the students of their programming progress. The intention is to survey the students again at the end of the module in an attempt to better understand their opinion of screencasts as a means of delivering feedback. The survey will attempt to elicit the students perceptions of the screencast feedback based on the categories of engagement, quality and quantity of feedback, helpfulness and comparison to written feedback.

REFERENCES

- [1] B. Law, "Screencasts : Enhancing Coursework Feedback for Game Programming Students," in The Twelfth International Multi-Conference on Computing in the Global Information Technology (ICCGI). Nice: The International Academy, Research and Industry Association (IARIA), 2017, pp. 17–21.
- [2] N. U. of Students (NUS), "The nation student survey," 2015, [retrieved: July, 2017]. [Online]. Available: <http://www.thestudentsurvey.com/>
- [3] S. A. Hope, "Making movies: The next big thing in feedback?" Bioscience Education, vol. 18, no. 1, 2011, pp. 1–14.
- [4] K. Haxton and D. McGarvey, "Screencasting as a means of providing timely, general feedback on assessment," New Directions, vol. 7, 2011, pp. 18–21.
- [5] N. U. of Students (NUS), "Nss 2015 national headlines," 2015, [retrieved: July, 2017]. [Online]. Available: <http://www.thestudentsurvey.com/>
- [6] R. Law, "Using screencasts to enhance coursework feedback for game programming students," in Proceedings of the 18th ACM conference on Innovation and technology in computer science education. ACM, 2013, pp. 329–329.
- [7] R. Thompson and M. J. Lee, "Talking with students through screencasting: Experimentations with video feedback to improve student learning," The Journal of Interactive Technology and Pedagogy, vol. 1, no. 1, 2012.
- [8] M. Robinson, B. Loch, and T. Croft, "Student perceptions of screencast feedback on mathematics assessment," International Journal of Research in Undergraduate Mathematics Education, vol. 1, no. 3, 2015, pp. 363–385.
- [9] K. Edwards, A.-F. Dujardin, and N. Williams, "Screencast feedback for essays on a distance learning ma in professional communication," Journal of Academic Writing, vol. 2, no. 1, 2012, pp. 95–126.
- [10] G. Stieglitz, "Screencasting: Informing students, shaping instruction," UAE Journal of Educational Technology and eLearning, vol. 4, no. 1, 2013, pp. 58–62.
- [11] S. Atfield-cutts and M. Coles, "Blended Feedback II : Video screen capture assessment feedback for individual students , as a matter of course , on an undergraduate computer programming unit," 2013. [Online]. Available: <http://eprints.bournemouth.ac.uk/23813/>
- [12] P. Race, "Using feedback to help students to learn," HEA, York, 2001.
- [13] K. Mathieson, "Exploring student perceptions of audiovisual feedback via screencasting in online courses," American Journal of Distance Education, vol. 26, no. 3, 2012, pp. 143–156.
- [14] F. Kiliçkaya, "Use of Screencasting for Delivering Lectures and Providing Feedback in Educational Contexts: Issues and Implications." Online Submission, 2016. [Online]. Available: <https://eric.ed.gov/?id=ED574888>
- [15] C. Evans, "Making sense of assessment feedback in higher education," Review of Educational Research, vol. 83, no. 1, 2013, pp. 70–120. [Online]. Available: <http://dx.doi.org/10.3102/0034654312474350>
- [16] J. Seror, "Show me! enhanced feedback through screencasting technology," TESL Canada Journal, vol. 30, no. 1, 2012, pp. 104–116.
- [17] M. Henderson and M. Phillips, "Video-based feedback on student assessment: scarily personal," Australasian Journal of Educational Technology, vol. 31, no. 1, 2015, pp. 51–66.
- [18] F. Harper, H. Green, and M. Fernandez-Toro, "Using screencasts in the teaching of modern languages: investigating the use of jing® in feedback on written assignments," The Language Learning Journal, 2015, pp. 1–18.
- [19] R. Galanos, W. Brand, S. Sridhara, M. Zamansky, and E. Zayas, "Technology we can't live without!: revisited," in Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education. ACM, 2017, pp. 659–660.
- [20] J. West and W. Turner, "Enhancing the assessment experience: improving student perceptions, engagement and understanding using online video feedback," Innovations in Education and Teaching International, 2015, pp. 1–11.
- [21] P. Marriott and L. K. Teoh, "Using screencasts to enhance assessment feedback: Students' perceptions and preferences," Accounting Education, vol. 21, 2012, pp. 583–598.
- [22] P. Mathisen, "Video feedback in higher education—a contribution to improving the quality of written feedback," Nordic Journal of Digital Literacy, vol. 7, no. 02, 2012, pp. 97–113.
- [23] P. OMalley, "Screencasting and a tablet pc—an indispensable technology combination for physical science teaching and feedback in higher and further education," in Aiming for excellence in STEM learning and teaching: Proceedings of the Higher Education Academy's First Annual Learning and Teaching STEM Conference, 2012.
- [24] C. Chewar and S. J. Matthews, "Lights, camera, action!: video deliverables for programming projects," Journal of Computing Sciences in Colleges, vol. 31, no. 3, 2016, pp. 8–17.
- [25] W. Sugar, A. Brown, and K. Luterbach, "Examining the anatomy of a screencast: Uncovering common elements and instructional strategies," International Review of Research in Open and Distance Learning, vol. 11, no. 3, 2010, pp. 1–20. [Online]. Available: <http://www.irrodl.org/index.php/irrodl/article/view/851>
- [26] P. Smith, "Screencasting as a means of enhancing the student learning experience," Learning and Teaching in Action, 2014, p. 59.
- [27] D. Wolff-Hilliard and B. Baethe, "Using digital and audio annotations to reinvent critical feedback with online adult students," International Journal for Professional Educators, 2014, p. 40.
- [28] S. Mohorovicic, "Creation and use of screencasts in higher education," in MIPRO, 2012 Proceedings of the 35th International Convention. IEEE, 2012, pp. 1293–1298.
- [29] S. Mohorovićić and E. Tijan, "Using Screencasts in Computer Programming Courses," in Proceedings of the 22nd EAEEIE Annual Conference, Maribor, 2011, pp. 220–225. [Online]. Available: http://www.eaeeie2011.uni-mb.si/eaeeie2011_submission_48.pdf
- [30] L. A. Jones, "Losing the red pen: Video grading feedback in distance and blended learning writing courses," Association Supporting Computer Users in Education Our Second Quarter Century of Resource Sharing, 2014, p. 54.
- [31] A. D. Ali, "Effectiveness of using screencast feedback on efl students writing and perception," English Language Teaching, vol. 9, no. 8, 2016, p. 106.

Revision Control and Automatic Documentation for the Development Numerical Models for Scientific Applications

Martin Zinner*, Karsten Rink[†], René Jäkel*, Kim Feldhoff*, Richard Grunzke*,
Thomas Fischer[†], Rui Song[‡], Marc Walther[§], Thomas Jejkal[¶], Olaf Kolditz^{†||}, Wolfgang E. Nagel*

* Center for Information Services and High Performance Computing (ZIH)

Technische Universität Dresden

Dresden, Germany

E-mail: {martin.zinner1, rene.jaekel, kim.feldhoff}@tu-dresden.de,
{richard.grunzke, wolfgang.nagel}@tu-dresden.de

[†] Department of Environmental Informatics

Helmholtz Centre for Environmental Research (UFZ)

Leipzig, Germany

E-mail: {karsten.rink, thomas.fischer, marc.walther, olaf.kolditz}@ufz.de

[‡] Technical Information Systems

Technische Universität Dresden

Dresden, Germany

E-mail: rui.song@tu-dresden.de

[§] Professorship of Contaminant Hydrology

Technische Universität Dresden

Dresden, Germany

E-mail: marc.walther@tu-dresden.de

[¶] Institute for Data Processing and Electronics

Karlsruhe Institute of Technology (KIT)

Karlsruhe, Germany

E-mail: thomas.jejkal@kit.edu

^{||} Professorship for Applied Environmental System Analysis

Technische Universität Dresden

Dresden, Germany

E-mail: olaf.kolditz@ufz.de

Abstract—As software becomes increasingly complex, automatic documentation of the development is becoming ever more important. In this paper, we present a novel, general strategy to build a revision control system for the development of numerical models for scientific applications. We set up a formal methodology of the strategy and show the consistency, correctness, and usefulness of the presented strategy to automatically generate a documentation for the evolution of the model. As a use case, the proposed system is employed for managing the development of hydrogeological models for simulating environmental phenomena within a research environment.

Keywords—Software development; Automatic generation of documentation; Revision control; Backup and Restore; Metadata; Improvement of research environment; Support of research process.

I. INTRODUCTION

In scientific applications, dedicated software packages are used to create numerical models for the simulation of physical phenomena [1]. In the scope of this paper we will focus on

environmental phenomena, such as the simulation of flooding, groundwater recharge or reactive transport using innovative numerical methods. Such simulations are crucial for solving major challenges in coming years, including the prediction of possible effects of climate change [2] [3], the development of water management schemes for (semi) arid regions [4] [5] or the reduction of groundwater contamination [6] [7].

The modeling process is usually a complete workflow, consisting of a number of recurring steps. To better understand the need for documentation and storing multiple versions of the same model, we would like to roughly outline the process:

- 1) *Data acquisition*: Relevant data sets required for setting up a model and parameterizing a process simulation are collected. For hydrogeological processes, this includes the digital elevation model (DEM), stratigraphic information from boreholes or other sources, production rates from wells, precipitation rates from climate stations, in- and outflow rates for the region of interest, etc.

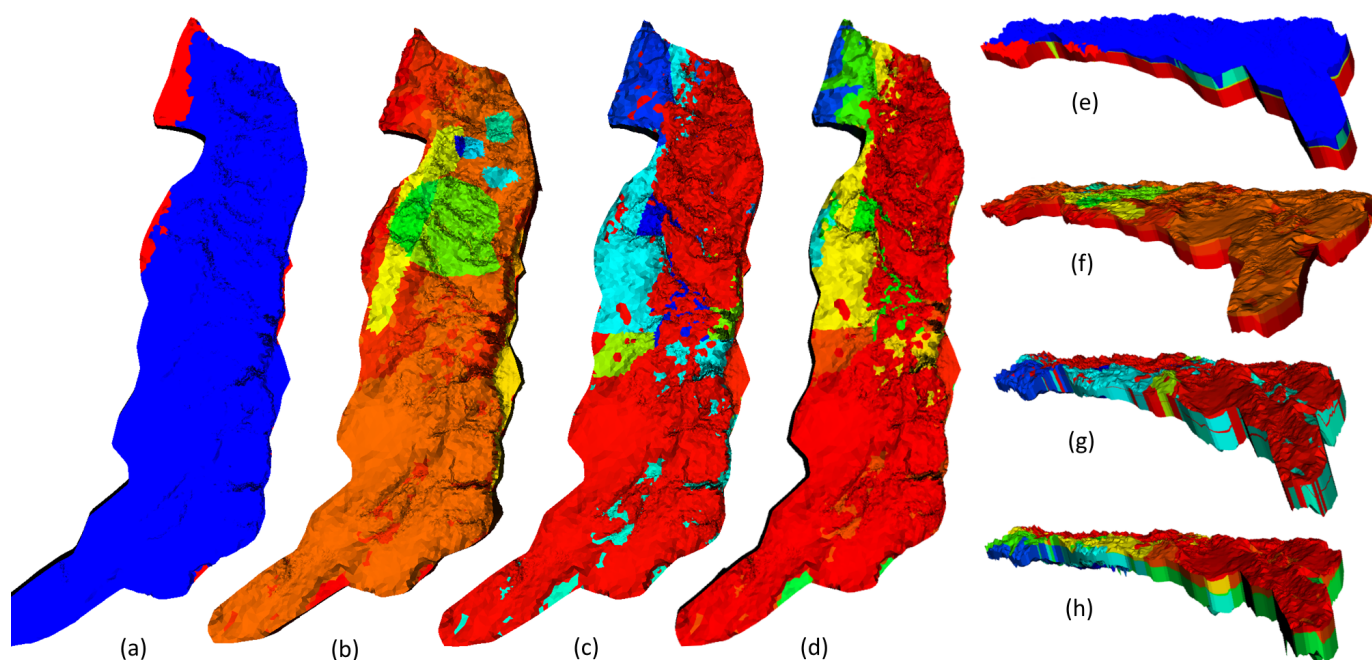


Figure 1. Example for the development of a numerical model over multiple iterations. All figures depict a numerical groundwater model for a region in the Middle East [8]. Figure (a) and (e) show the initial model from January 2011. Figure (b) and (f) depict that state of the model in August 2011 when more information on the surface had been added. Figure (c) and (g) show the complex state at December 2012 when also additional subsurface information had been integrated. Finally, Figure (d) and (h) show the final state of the model in April 2013 after a number of simplifications. Figure (a)–(d) show the top of region of interest, while Figure (e)–(h) show an isometric view such that stratigraphic information is visible.

- 2) *Data integration:* Information is usually collected from different sources and data sets have been acquired using various measurement devices (e.g., remote sensing data from satellites, sensor data, manually created logs) When aggregating the information, artifacts in data and inconsistencies between data sets need to be removed. This includes fairly simple tasks, such as projecting all data sets into the same geographic coordinate system, but also challenging work, such as dealing with missing or conflicting data [9].
- 3) *Model Generation:* Information from the input data sets is used to create a numerical model. For instance, DEM and borehole information are used to create a finite element mesh of the subsurface region of interest, source- and sink terms have to be integrated into the mesh, precipitation and outflow information are used to create boundary conditions, and mesh elements need to be parameterized based geohydrological parameters [10].
- 4) *Process Simulation:* Time stepping scheme, non-linear solver type, pre-conditioner and linear solver for the numerical schemes need to be selected and parameterized. For HPC-Applications, a domain decomposition needs to be performed for the the subsurface mesh. At this point, one or multiple runs of the actual simulation are done [11]
- 5) *Validation / Visualization:* Simulation results are visualized and checked for plausibility. Results are compared to actual numbers, for example from observation wells, outflow measurements or using other means of validation [12].

In reality, the above workflow will not be executed as linear

as suggested above. Often, it is not obvious exactly, which data sets are necessary to run a meaningful simulation. Precise data sets are often hard to come by and need to be requested from state offices or bought from commercial services. The best practice is to set up one or more initial models using data that is available at the time to get a first prototype and see potential problems during simulation or when comparing results to actual validation data. In addition, researchers usually want to create a model that is as complex as necessary but as simple as possible. Complex models tend to be more precise but have more degrees of freedom: boundary conditions and (coupled) processes are hard to parameterize and numerically challenging, the run time is usually (much) longer and problems occurring due to the structure of the model are harder to track down. In contrast, simple models might not be able to represent the region of interest or the simulated process adequately and the correctness or precision of simulation results may be insufficient. Unfortunately, the modeling process itself in general is not transparent and traceable and often poorly documented. A typical model – consisting of a set of parameter files – is developed over many weeks or months (see Figure 1). Usually a large number of revisions are necessary to update and refine the model, such that the simulation represents the natural process as realistically and plausibly as possible. Examples for reasons to adjust the finite element mesh representing a subsurface model domain include changes to element size used to either allow for an adequate representation of the processes of interest (e.g., groundwater flow, heat conduction, dispersion of chemical compounds), integration of additional datasets (e.g., river / stream networks, wells, distribution of soil types) or availability of more precise measurements for data already integrated, re-meshing due to numerically difficult configurations, or

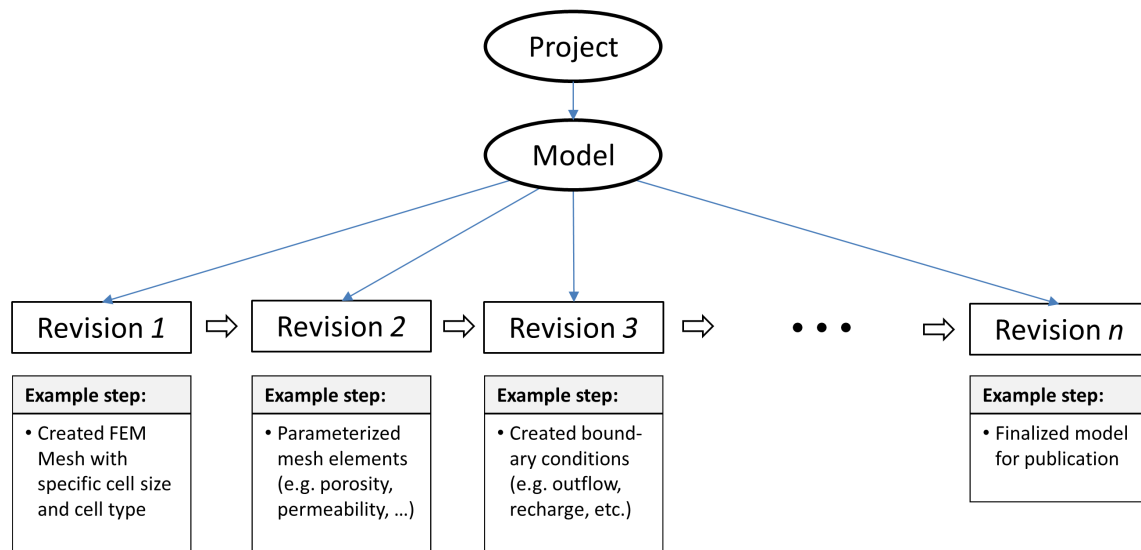


Figure 2. Model development over several revisions

information reduction when the model has become too complex. Other parts of the model, such as boundary conditions or the numerical configuration might also change for different reasons. Especially with multiple researchers working on one model, it is not hard to imagine that it can become difficult to keep track of changes and the reasons why certain aspects of a model have been adjusted; especially when models are developed or maintained over a period of multiple years. It can also become difficult to restore a certain previous state of a model after a series of changes by one or more scientists over a given period of time.

In this paper, we address these particular challenges. Specifically, we present a revision control system, which in addition to the backup/restore functionality tracks the changes in each modeling step, thus generating an internal documentation of the evolution of the model.

A. Technical Challenges and Objectives

The basic concept for the simulation of a certain phenomenon in a given region is a model. As shown in the previous section, this model is developed over several iterative steps. In the scope of this paper, we will refer to these steps as “revisions”, compare Figure 2. As mentioned, the first setup of the model is often used to get an overview over existing data and to get familiar both with the region of interest as well as the basic behavior of phenomena within that region. At this stage, potential problems, missing data or specific numerical requirements might already become apparent. After creating the first revision (as well as after each subsequent modeling step) a simulation is run, using the model. Depending on the result of the process simulation, further revisions will try to solve these issues by addressing the shortcomings of the simulation result. Typical follow-up steps include adding refined or previously missing data, adjusting or refining the finite element mesh, changing process parameterizations or numerical schemes, etc. (see Figure 3).

The framework for revision control for environmental model development is being implemented at the Helmholtz-Centre

for Environmental Research (UFZ) [13] using the Karlsruhe Institute of Technology Data Manager (KIT DM) [14] as a software framework for creating and maintaining repositories for research data.

The Metadata Management for Applied Sciences (MASi) [15] research data management service is currently being prepared for production at the Center for Information Services and High Performance Computing (ZIH) at Technische Universität Dresden. It utilizes the advanced KIT DM framework to provide a service that enables the metadata-driven management of data from arbitrary research communities. This includes automating as many processes as possible including metadata generation and data pre-processing.

The current solution, utilized at UFZ, is completely file based and it is usually stored locally on the laptop of each scientist.

Depending on the complexity of the model and the phenomena that need to be simulated, the number of parameter files varies between three and several hundred, with each file up to several megabytes. The minimum configuration for the OpenGeoSys simulation software [11] requires a finite element mesh, geometric information to specify spatial conditions, as well as a project file containing all process-based information and numerical parameterizations. However, for complex case studies, additional files may become part of the model, for instance to represent boundary conditions. Examples include weather radar data (typically one file per timestep), data from observation wells (typically one time series per well), geometries of changing conditions in the model domain such as advancing/receding coastlines during floods/droughts (one file per timestep). The changes from one revision of the model to the next can be very small, e.g., when one parameter value changed in an input file. However, the changes can also become major, for example when a geometric constraint is updated, which in turn requires re-meshing the model domain and adjusting associated boundary conditions and possibly even the precise type of process that is used because the domain that

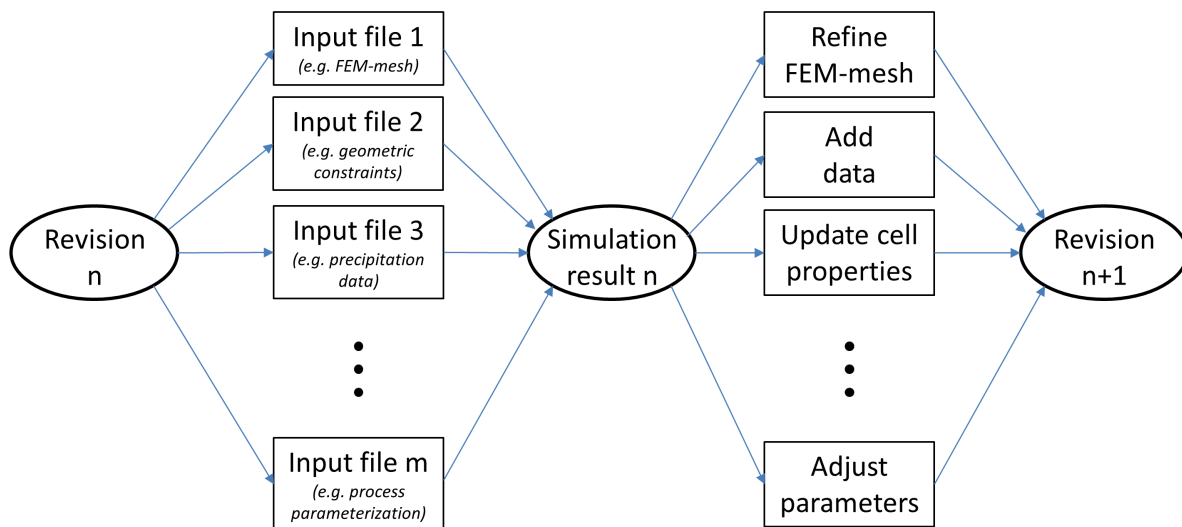


Figure 3. File-based view of revisions and simulation as part of the model development process. Here, a revisions is represented by a collection of input files. Running a simulation will give a set of output files. If simulation results are not satisfactory given existing validation measurements or the researcher's experience, or if the simulation does not converge at all, changes to the model will be made based on the result. Examples include refining the mesh (e.g., if the simulation started to oscillate), adding data (such as a river geometry to use as an additional boundary condition), update cell properties (e.g., adjust permeability of stratigraphic layers so groundwater flow will behave differently) or adjust the parameterization of the numerical process (e.g., choose smaller time steps if the simulation did not converge).

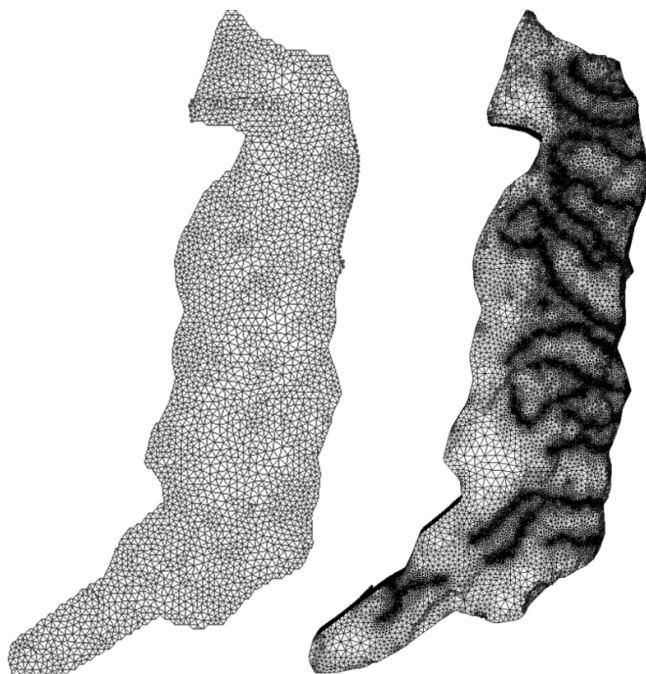


Figure 4. Example for changes of a finite element mesh: The left-hand side depicts the homogenous mesh created for the first iteration of the modeling process, the right hand side the final adaptive mesh, which is refined towards rivers and wells [8].

had previously been considered saturated is now unsaturated. Figure 4 shows an example of changes made to a finite element mesh during the development of a model.

The main deficiencies for researchers working with the current file-based solution are:

- 1) No overview of the development of the model (especially after handling the model for a long time)
- 2) Difficulty to trace parameter changes and the reasons for those changes
- 3) No implicit or explicit documentation of the changes
- 4) Each user stores the data on his laptop at his own discretion
- 5) Data is lost if hard disk crashes and there is no backup
- 6) Joint working on the same model is cumbersome

The benefits of the new framework will include the advantages of a classical revision control system (like Git [16], or Apache Subversion [17]), in particular:

- 1) Uniform, central, and consistent storage of the individual modeling steps a) each scientist will be able to view the simulation data he is entitled to b) backup functionality if the data is lost,
- 2) possibility to track and analyze / evaluate the changes,
- 3) data is still available if the PhD student leaves the company,
- 4) shared access of the latest development of the model.

A revision is defined as the state of the components already persisted and accessible by a unique identifier. Thus, the content of the components of a revision cannot be altered any more. The current set of the components, which can be actively changed is called the *working set*.

The main objectives we focus on, to achieve our scope, are:

- 1) Central persistent storage of the model to include all the modeling steps and the management of the revisions.

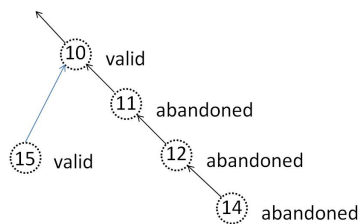


Figure 5. Tree structure of the development of the model

- 2) Design and development of a metadata repository regarding a) revision control and b) the changes of the parameter files between subsequent revisions. Additionally, information regarding parameter values, simulation software, etc. can be persisted.
- 3) An efficient and disk space saving strategy, such that a specific parameter file is stored only if its content has been modified.
- 4) Generation of an internal documentation of the model development, such that it can be easily understood and reconstructed.

It is out of scope of this research to persistently store the results of the simulation. If necessary, it can be generated again or a direct storage strategy could be used. Storing the results of the simulation together with the parameter files would leverage our sophisticated storage strategy, since the size of the parameter files is in the range of megabytes, where the size of the simulation files is in the range of gigabytes. The storage of the simulation results is only meaningful if it takes too long to newly generate the results. The model development is stored in a tree structure, such that each node (revision) has a unique link to its predecessor, see Figure 5. The tree structure is necessary to be able to identify modeling steps where the results of the simulation are not promising, and thus this revision is not pursued further (termed *abandoned*). In this case, the development of the model is continued from a previous revision (termed *active*), thus performing a rollback on the evolution of the model and creating a new branch in the version control tree structure.

Usually, metadata is defined as data about data. Metadata files can be generated automatically or they can be set up manually. The *flow configuration file* is the main metadata file and it is generated automatically during the evolution process of the model and contains the basic information regarding the revision control system, which is necessary to generate the internal documentation of the evolution of the model, i.e.,

- a) model name;
- b) predecessors and the current revision;
- c) cryptographic hash value and status of the parameter files;
- d) parameter change information in condensed form, etc.

The main metadata file sustains the possibility to automatically capture, track, analyze, and evaluate the changes in each modeling step.

Additionally, users can define their own metadata files, which can be created for the whole case study or for a specific revision and could contain additional information regarding a) name of the project; b) model area; c) modeled process; d) software used, including the version; e) contact person; f) source reference regarding the applied methods and data used; g) utilization rights, etc.

Besides documentation, metadata also allows for easy identification of the uploaded data. Search for specific values (e.g., model name, author, etc.) over all metadata elements can be performed for example by using ElasticSearch [18].

B. Outline

The structure of the paper is as follows: In Section II, we give a short overview over the state of art and detail some differences of our approach, both in concept and realization; in Section III, we demonstrate our novel strategy, which is used for the revision control system to generate the implicit documentation of the evolution of the model; in Section IV, we augment the classical pseudo code presentation of the algorithms to a formal, mathematical description of our selective backup strategy and show the consistency and correctness of the backup and restore functionalities. We present the software implementing the formal description and its application to a use case of the UFZ in Section V. Finally, we conclude our work and give an outlook for future research and development in Section VI.

II. RELATED WORK

The concept of revision control systems (RCS) is not new (see Tichy [19]). The task of the RCS as defined by Tichy is version control, i.e., keeping software systems consisting of many versions and configurations well organized. The concept of a revision is similar to our approach, an ancestral tree is used for storing revisions. The major difference is that – as set up by Tichy – each object (like a file) has his own revision tree, whereas we follow an overarching concept, such that files may remain unchanged between revisions. Furthermore, the evolvement of the revision is linear, but it can use *side branches*, for example one for the productive version and one for the development [19].

Löh et al. [20] present a formal model to reason about version control, in particular modeling repositories as a multiset of patches. Patches abstract over the data on which they operate, making the framework equally suited for version control from highly-structured XML to blobs of bits. The mathematical definition of patches and repositories enable Löh et al. to reason about complicated issues, such as conflicts and conflicts resolution. The main application field that Löh et al. targets is the distributed (software) development with its challenges regarding the complex operations on the repositories, such as merging branches or resolving conflicts. They introduce a precise, mathematical description of the version control system to accurately predict when conflicts may arise and how they may be resolved.

Our mathematical model is not based on the work of Löh et al., it has been developed from scratch to enable the characterization of the selective backup strategy.

The possibility to use metadata, such as the patch's author, time of creation, or some form of documentation is shortly discussed in [20]. Details are left to the designers of a specific revision control system. Also the concept of reverting changes, i.e., the ability to return to a previous version by undoing a modification that later turns out to be undesired, is discussed from a theoretical point of view.

As stated in [21] there are some basic goals of a versioning system, such that:

- 1) People are able to work simultaneously, not serially.
- 2) When people are working at the same time, their changes do not conflict with each other.

These two goals do not apply in our case. Formally, users can work simultaneously, making changes independently, but for a simulation they need all the parameter files. The classical use case, such that a programmer changes the internal specification of a module without changing the external interface is not applicable in our case, each change in a parameter file leads to different simulation results. Unfortunately, the usual versioning systems do not support our advanced requirements regarding usage of metadata and enhanced automatic documentation generation of the evolution of the model.

The automatic generation of documentation has also been the scope of intense academic and industrial research. It has been recognized that the importance of good documentation is critical for user acceptance [22]. Jesus describes in [23] a paradigm for automatic documentation generation based on a set of rules that, applied to the models obtained as result of the analysis and design phases, gives an hypertext network describing those models. On the contrary, our approach has the advantage that the algorithm that is used for the selective backup strategy also delivers the data for the automatic documentation. PLANDoc [24] documents the activity – of planning engineers – by taking as input a trace of the engineer's interaction with a network planning tool. Similarly, in [25] Alida, an approach for fully automatic documentation of data analysis procedures, is presented. During analysis, all operations on data are registered. Subsequently, these data are made explicit in XML graph representation, yielding a suitable base for visual and analytic inspection. The high level approach in Alida – using the information generated during the production process to automatically create the documentation – is similar to ours.

As a final note, we looked very carefully at the existing revision control systems, both at the academic and the commercial ones, and found no adequate revision control system, worth to adapt to fulfill our needs towards a system, which automatically supports and tracks the evolution of the model during its development process.

III. SELECTIVE BACKUP STRATEGY

The aim of our revision control system is to provide an enhanced backup strategy, termed *selective backup strategy*, such that only the components of the working set that have been modified are considered for backup, see Figure 6. This is an enhancement of the usual incremental backup strategies, storing a particular modification of a file only once, in order to provide the framework to generate the metadata regarding

the modifications and accordingly to generate the implicit documentation of the model.

A correspondent *selective restore strategy* is used, i.e., the latest versions of all components are downloaded, such that at the end the recent version of the model is assembled out of the historical backups.

A. Backup

Only part of the current working set is uploaded into the data repository, and the uploaded information cannot be altered or removed later. According to our selective backup strategy a) for the first revision: all components are uploaded (see Figure 6 left side); b) for the subsequent revisions: only the components that have been modified are uploaded (see Figure 6 right side).

B. Restore

It will be possible to download all relevant information regarding a specific revision (including parameter files and metadata files). This requires identifying and downloading the full set of components necessary to run a simulation. The required information is stored in the main metadata file during the backup process. Hence, the main metadata file stores information regarding all files that have been uploaded including the unique identification of the uploaded object and the cryptographic hash values of the respective files.

1) *Full Restore*: The full restore should be applied if files have been lost, or the development of the model is intended to be pursued by other users, etc. The full restore retrieves the whole set of parameter files, such that a simulation can be done on the restored system.

2) *Revision Restore*: This functionality restores the files corresponding to a (previous) revision and permits to continue the simulation corresponding from that revision. This method enables the tree structure of the revision history. The corresponding information is retrieved from/written to the main metadata file.

C. Flow Configuration

We present now some implementation details. The relevant information for the functioning of the selective backup and the corresponding restore strategy is stored / updated automatically – using XML – in the flow configuration file.

This file stores general information as: a) short name of the model; b) the number of the last revision; c) the object id under which the files belonging to that revision were uploaded; d) additional information in order to identify the project, the revision, etc. A simple example is given in Figure 7. Additionally, general information regarding the revision history such as: a) the revision number; b) the object id of the uploaded object; c) the predecessor (parent revision); the total number of files versus the number of files, which have been changed and in consequence uploaded, etc., as given in Figure 8. File and revision specific information are also tracked, such that for each file the revision where the file has been changed and the corresponding cryptographic values are tracked. This way, it is ensured that a specific state of a file is stored only once and that the revision under which this file has been stored can unambiguously be identified, see Figure 9 for an example.

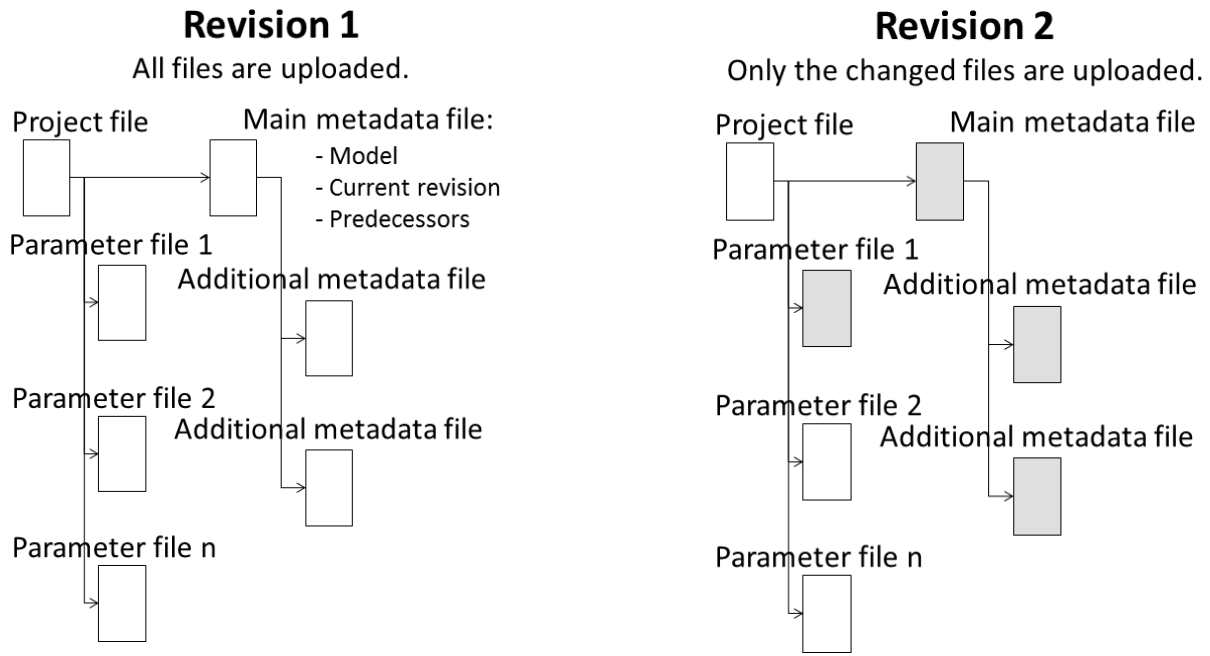


Figure 6. Selective backup strategy. Uploaded files at first (left) and second revision (right).

```

1 <GeneralConfiguration>
2   <ModelShortName>cube_1e0_neumann</ModelShortName>
3   <LastRevisionNr>25</LastRevisionNr>
4   <LastDigitalObjectID>3ccafed6-cf0d-486d-bbe3-
   edff4159f6c5</LastDigitalObjectID>
5   <LastNotes>cube_1e0_neumann / Incr. / It.Nr.: 25 /
   Standard Incremental Upload</LastNotes>
6 </GeneralConfiguration>

```

Figure 7. Excerpt 1 of example configuration file

```

1 <Revision>
2   <RevisionNr>7</RevisionNr>
3   <DigitalObjectID>8ce20b42-c15f-427c-ac1a-d575295f5412</
   DigitalObjectID>
4   <ParentRevisionNr>3</ParentRevisionNr>
5   <TotalNrFiles>20</TotalNrFiles>
6   <NrFilesUploaded>1</NrFilesUploaded>
7   <Notes>cube_1e0_neumann / Incr. / It.Nr.: 2 / For Testing
   Upload and Download</Notes>
8 </Revision>

```

Figure 8. Excerpt 2 of example configuration file

```

1 <FileCharacteristics>
2   <FileName>cube_1x1x1.gml</FileName>
3   <FileLastRevision>43</FileLastRevision>
4   <FileHistory>
5     <FileRevision>
6       <StorageRevisionNr>1</StorageRevisionNr>
7       <StorageDigitalObjectID>8aac5970-a366-49ce-a052
       -6c8f82ced85c</Storage\~Digital\~ObjectID>
8       <FileSize>1622</FileSize>
9       <FileCreationTime>2016-08-18T08:25:33Z</
       FileCreationTime>
10      <FileLastAccessTime>2016-08-23T15:59:55Z</
       FileLastAccessTime>
11      <FileLastModifiedTime>2016-08-18T08:25:33Z</
       FileLastModifiedTime>
12      <FileCryptoIDMD5>66
       a9800e8b02d85001cdd13930b85ea3</
       FileCryptoIDMD5>
13      <FileCryptoIDSHA1>5
       c942ba146b41e38262f23ed350e214c757d8803</
       FileCrypto\~IDSHA1>
14    </FileRevision>

```

Figure 9. Excerpt 3 of example configuration file

D. Accurate versioning

Based on the architecture of the system, the selective backup strategy corresponds to a centralized revision control system, i.e., there is a central revision number – in our case the modeling step –, such that the version of each file is tied to this central revision number. In contrast, revision control systems like Git [16] are decentralized, i.e., generally, users maintain the versioning of their part, without affecting the overall release number. In our case, this centralized approach is of crucial importance, since small changes in one parameter file can substantially affect the outcome of the simulation. The selective backup strategy enables a paradigm change in the theory and practice of (centralized) revision control systems, it enables

an accurate tracking of the changes during each revision on file level including the identification of the effective version of each file. This means especially that in contrast to Git and SVN [17], during revision change, each file is compared to previous versions and either assigned a new version number or – if possible – reassigned a previous one. Such a distinction is not absolutely necessary, for example during software development (main application field for Git and SVN), but it is of crucial importance for pursuing the exact model development.

We illustrate now the selective backup strategy by means of the example as delineated in Table I. Let $\{F_1, F_2, F_3, \dots, F_n\}$ be files comprising a numerical model and $\{R_1, R_2, R_3, \dots, R_m\}$ the revisions to adjust the model for a successful simulation of a process. According to the architecture we use, the number of revisions is greater than the number of the files involved, i.e.,

$n < m$. We number the versions of a specific file continuously in the order they were generated, starting with 1. We notate by an upper index the revision during which the version was generated, i.e., for the file F_1 the version $V_4^{R_6}$ represents the fourth version generated during revision R_6 . Not all files are necessarily updated with a revision. For instance, file F_1 is unchanged during revisions R_4, R_5 , thus the model keeps using the file version $V_3^{R_3}$. In contrast, file F_2 is modified in each revision. However, during revision R_5 , the file is reverted to a previous state such that its version is equal to $V_2^{R_2}$ and, instead of storing a duplicate, the previous copy of the file is used.

Using revision systems like Git or SVN, a new version of the file F_2 would be created. Instead, our algorithm, using the selective backup strategy, verifies if a version with new content has been created or the respective content has already been used before.

The use of the selective backup strategy is not restricted to the model development for scientific application, but can be applied everywhere where a centralized revision control system is used. Its intended target are applications, which need to track the effective version of the files, potentially related to revision numbers.

IV. THE FORMAL MODEL

We introduce a mathematical model in order to use the advantages of the rigor of a formal approach over the inaccuracy and the incompleteness of natural languages. We augment the classical pseudo code presentation of the algorithms to a formal, mathematical description and show the consistency and correctness of the backup and restore functionalities.

A. Notations

We use a calligraphic font to denote the index sets. We denote by $\mathcal{C} := \{C_i \mid i \in \mathcal{C} \text{ and } C_i \text{ is a component}\}$ the finite set of the components, i.e., the disjunct union of the parameter files and the metadata files. Let S be an arbitrary set. We notate by $\mathcal{P}(S)$ the power set of S , i.e., the set of all subsets of S , including the empty set and S itself. By $\text{card}(S)$ we notate the cardinality of S . Let $n \in \mathbb{N}$ and let $f : X \rightarrow X$ be a function. Finally, we denote by $f^n : X \rightarrow X$ the function obtained by composing f with itself n times, i.e., $f^0 := \text{id}_X$ and $f^{n+1} := f^n \circ f$.

B. Introducing Components and Revisions

Some components – at least one, but not necessary all – are modified, then a simulation is performed. We call this state of the components a *revision*. Each revision is backed up to a persistent storage. We have in a natural way a total ordering $<$ on the set of the revisions considering the order they were generated. We denote by \mathcal{R} the ordered set of the revisions, i.e., $\mathcal{R} := \{R_i \mid i \in \mathcal{R} \text{ and } R_i \text{ is a revision}\}$. Let $m := \text{card}(\mathcal{R})$ be the number of revisions. In order to keep the notations straightforward we set $\mathcal{R} := \{1, 2, 3, \dots, m\}$, such that $\forall i \in \mathcal{R} \setminus \{m\} : R_i < R_{i+1}$. We denote by $C_k^{(i)}$ the component C_k having the state at revision R_i , therefore, we denote by $\mathfrak{R}_i := \{C_k^{(i)} \mid k \in \mathcal{C}\}$ the set of the components having the state corresponding to revision R_i .

We denote by \mathfrak{R} the matrix of the evolution of the model, hence $\mathfrak{R} := \{C_k^{(i)} \mid k \in \mathcal{C} \text{ and } i \in \mathcal{R}\}$. Therefore, \mathfrak{R} contains the history of the content changes of the components during the evolution process of the model.

Let *HASH* be the set of all the hash values. We define the content of a component $C_k \in \mathcal{C}$ corresponding to a revision R_i formally as the function:

Definition IV.1 (Content of components) We set

$$\text{CONT}: \mathfrak{R} \rightarrow \text{HASH},$$

$$C_k^{(i)} \mapsto \text{CONT}(C_k^{(i)}) := \text{hash value of } C_k^{(i)}.$$

Remark IV.1 Let $k \in \mathcal{C}$, let $i, j \in \mathcal{R}$, such that $i \neq j$. In order to track the change process during the evolution process of the model, we are interested only in comparing the content of the same component at different revisions (i.e., the content of $C_k^{(i)}$ versus the content of $C_k^{(j)}$). ■

Definition IV.2 (Origin of a revision) Let $R, Q \in \mathcal{R}$. We say that Q is the origin of R , notated by $Q = \text{ORIGIN}(R)$, if and only if the revision R has been obtained by direct modification of the content of the components having the state at revision Q . For formal reasons we define $\text{ORIGIN}(R_1) := R_1$.

Remark IV.2 Let $R \in \mathcal{R}$ arbitrarily chosen. Then there exists a unique $Q \in \mathcal{R}$, such that $Q = \text{ORIGIN}(R)$. This is a direct consequence of the definition above (see Definition IV.2). ■

Let $m = \text{card}(\mathcal{R})$, such that $m \geq 1$. We define the predecessor and the successor of a revision formally as:

Definition IV.3 (Predecessor of a revision) We set

$$\text{PRED}: \mathcal{R} \rightarrow \mathcal{R},$$

$$R \mapsto \text{PRED}(R) := \text{ORIGIN}(R).$$

Definition IV.4 (Successor of a revision) We set

$$\text{SUCC}: \mathcal{R} \rightarrow \mathcal{P}(\mathcal{R}),$$

$$R \mapsto \text{SUCC}(R) := \{Q \in \mathcal{R} \mid R = \text{ORIGIN}(Q)\}.$$

Remark IV.3 $\forall R \in \mathcal{R} \Rightarrow \text{PRED}(R)$ is unequivocally determined (see Remark IV.2), in contrast, there exists to a revision $R \in \mathcal{R}$ a subset J of \mathcal{R} , such that $\text{SUCC}(R) = \{R_j \mid j \in J\}$. ■

Unfortunately, the structure of the evolution of the model is not linear. If for any reason the evolution of the model is in impasse, then the development of the model is not continued from the latest revision, but a previous revision is taken as a starting point. The revision, which led to the impasse is not pursued any more (i.e., it is *abandoned*). On the other side, a revision is *active* if it is part of the successful completion of the model. Formally, we define the status of a revision as follows:

Definition IV.5 (Status of a revision) Let $m := \text{card}(\mathcal{R})$ the number of revisions. We set

$$\text{STATUS}: \mathcal{R} \rightarrow \{\text{active}, \text{abandoned}\},$$

$$R \mapsto \text{STATUS}(R) := \begin{cases} \text{active} & \text{if } \exists n \geq 0 : \\ & R = \text{PRED}^n(R_m), \\ \text{abandoned} & \text{otherwise.} \end{cases}$$

Table I. Example for the selective backup strategy.

	R_1	R_2	R_3	R_4	R_5	R_6	\dots	R_m
F_1	$V_1^{R_1}$	$V_2^{R_2}$	$V_3^{R_3}$	$V_3^{R_3}$	$V_3^{R_3}$	$V_4^{R_6}$	\dots	$V_{m_1}^{R_{m_1 k_1}}$
F_2	$V_1^{R_1}$	$V_2^{R_2}$	$V_3^{R_3}$	$V_4^{R_4}$	$V_2^{R_2}$	$V_5^{R_6}$	\dots	$V_{m_2}^{R_{m_2 k_2}}$
F_3	$V_1^{R_1}$	$V_1^{R_1}$	$V_2^{R_3}$	$V_3^{R_3}$	$V_1^{R_1}$	$V_4^{R_6}$	\dots	$V_{m_3}^{R_{m_3 k_3}}$
\vdots	\vdots	\vdots	\vdots	\vdots	\vdots	\vdots		\vdots
F_n	$V_1^{R_1}$	$V_1^{R_1}$	$V_1^{R_1}$	$V_1^{R_1}$	$V_2^{R_5}$	$V_2^{R_5}$	\dots	$V_{m_n}^{R_{m_n k_n}}$

Informally, the predecessor of a component $C_k^{(i)}$ is the component $C_k^{(j)}$, such that R_j was the latest revision where the component C_k has been changed. Formally, we model the successor and predecessor of a component C_k during the revision process as a function.

Let $i \in \mathcal{R}$ and $k \in \mathcal{C}$ arbitrarily chosen. Set $A(i, k) := \max\{l \in \mathcal{R} : l < i \text{ and } \text{CONT}(C_k^{(l)}) \neq \text{CONT}(C_k^{(i)})\}$ then

Definition IV.6 (Predecessor of a component) We set

$\text{PRED}: \mathcal{R} \rightarrow \mathcal{R}$,

$$C_k^{(i)} \mapsto \text{PRED}(C_k^{(i)}) := \begin{cases} C_k^{(i)} & \text{if } (i = 1), \\ C_k^{(A(i, k))} & \text{if } (i > 1) \\ & \text{and } A(i, k) \text{ exists,} \\ C_k^{(1)} & \text{otherwise.} \end{cases}$$

Definition IV.7 (Successor of a component) We set

$\text{SUCC}: \mathcal{R} \rightarrow \mathcal{P}(\mathcal{R})$,

$$C_k^{(i)} \mapsto \text{SUCC}(C_k^{(i)}) := \{C_k^{(j)} \in \mathcal{R} \mid C_k^{(i)} = \text{PRED}(C_k^{(j)})\}.$$

Remark IV.4 The predecessor of $C_k^{(i)}$ is uniquely determined. This follows directly from the definition above. In contrast, the successor of $C_k^{(i)}$ is not necessary unique, but there exists a unique $R_j \in \mathcal{R}$, such that $C_k^{(j)} \in \text{SUCC}(R_k^{(i)})$ and $\text{STATUS}(R_j)$ is active. Similar considerations also hold for revisions. ■

Proposition IV.1 (Existence and uniqueness) Let $R \in \mathcal{R}$, such that $\text{STATUS}(R) = \text{active}$. If $\text{SUCC}(R) \neq \emptyset$ then there exists a unique $Q \in \mathcal{R}$, such that $Q \in \text{SUCC}(R)$ and $\text{STATUS}(Q) = \text{active}$.

Hint The existence and the uniqueness follows directly from the definition of the status of a revision (see Definition IV.5) and the uniqueness of the predecessor (see Remark IV.3). ■

We are now able to formulate our strategy to generate the successive revisions.

Lemma IV.1 (Linearity) Let $m = \text{card}(\mathcal{R})$. Then there exists a unique subset \mathcal{R}' of \mathcal{R} with $\mathcal{R}' = \{R_1, R_{i_1}, R_{i_2}, R_{i_3}, \dots, R_{i_l}, R_m\}$, such that $R_{i_1} \in \text{SUCC}(R_1)$ and $\forall i_k : i_1 \leq i_k < i_l \Rightarrow R_{i_{k+1}} \in \text{SUCC}(R_{i_k})$ and $R_m \in \text{SUCC}(R_{i_l})$ and $\forall R \in \mathcal{R}' : \text{STATUS}(R) = \text{active}$ and $\forall R \in \mathcal{R} \setminus \mathcal{R}' : \text{STATUS}(R) = \text{abandoned}$.

Hint It is a direct consequence of the uniqueness of active successors (see Proposition IV.1). ■

Corollary IV.1 The sequence of the active revisions is linear. ■

Let $i \in \mathcal{R}$ arbitrarily chosen, a component can have at the revision R_i two statuses *modified* and *preserved*, the value *modified* means that the component has been modified during the revision R_i , in contrast *preserved* means that the component remained unchanged at revision R_i . More formally, we define the function:

Definition IV.8 (Status of a component) We set

$\text{STATUS}: \mathcal{R} \rightarrow \{\text{modified}, \text{preserved}\}$,

$$C_k^{(i)} \mapsto \text{STATUS}(C_k^{(i)}) := \begin{cases} \text{modified} & \text{if } (i = 1), \\ \text{modified} & \text{if condition 2 holds,} \\ \text{preserved} & \text{otherwise.} \end{cases}$$

with condition 2: $\forall j \in \mathcal{R}$ with $1 \leq j < i : \text{CONT}(C_k^{(j)}) \neq \text{CONT}(C_k^{(i)})$.

Remark IV.5 From a formal point of view, all components corresponding to the first revision are considered modified. For the subsequent revisions only the components whose content has been altered are considered modified. ■

C. Backing up a revision

Based on the values of the *STATUS* function, we define the upload strategy. We are interested to upload only those components, which have been modified since the latest revision and the current state has not been uploaded previously.

Definition IV.9 (Upload of a component) We set

$\text{UPLOAD}: \mathcal{R} \rightarrow \{\text{yes}, \text{no}\}$,

$$C_k^{(i)} \mapsto \text{UPLOAD}(C_k^{(i)}) := \begin{cases} \text{yes} & \text{if condition 1 holds,} \\ \text{no} & \text{otherwise.} \end{cases}$$

with condition 1: $\text{STATUS}(C_k^{(i)}) = \text{modified}$.

Remark IV.6 This means especially that $C_k^{(i)}$ will be uploaded if and only if it has been modified and its content is different from the content of all its predecessors. ■

We will define now a function in order to model the backup process of an entire revision. As we will see, only those components will be backed up at a specific revision R_i , which have been modified at this revision.

Definition IV.10 (Backup of a revision) We set

$$\begin{aligned} \text{BACKUP}: \mathcal{R} &\rightarrow \mathcal{R}, \\ R_i &\mapsto \text{BACKUP}(R_i) \\ &:= \{C_k^{(i)} \mid k \in \mathcal{C}, \text{ such that } \text{UPLOAD}(C_k^{(i)}) = \text{yes}\}. \end{aligned}$$

Remark IV.7 This means especially that the components that have not been changed at revision R_i are not included in the backup of the revision R_i , this is the quintessence of the selective backup strategy. ■

In order to be able to model the download and restore process, we need to do some additional analysis. Those opposite functions cannot be defined straightforwardly as the reverse function of *BACKUP* and *UPLOAD*, since after the restore is fulfilled, all the relevant components must be available, not only those persisted at the corresponding revision.

In order to have all the relevant information for the restore process, we build during the evolution of the model a matrix $(\text{INF}_k^{(i)})_{k \in \mathcal{C}, i \in \mathcal{R}}$, such that this matrix contains the information relevant for the download and restore operations. This information contains the content of the components, such that comparisons can be done and relate it to the previous backups.

Hence, the matrix $(\text{INF}_k^{(i)})_{k \in \mathcal{C}, i \in \mathcal{R}}$ contains at least the information regarding the revision at which the component was physically stored, such that it can be retrieved from there and additional information regarding the content (hash value) of the components.

Formally, we define $(\text{INF}_k^{(i)})_{k \in \mathcal{C}, i \in \mathcal{R}}$ as a function:

Definition IV.11 (Component upload meta inf) We set

$$\begin{aligned} \text{INF}: \mathcal{R} \times \mathcal{C} &\rightarrow \mathcal{R} \times \text{HASH}, \\ (i, k) &\mapsto \text{INF}(i, k) := (j, v) \\ &\text{if } (C_k^{(i)} \in \text{BACKUP}(R_j) \text{ and } \text{CONT}(C_k^{(i)}) = v). \end{aligned}$$

Remark IV.8 This means especially that a component $C_k^{(i)}$ having the hash value $= v$ has been backed up at revision R_j and $j \in \mathcal{R}$ is the lowest index number, such that $\text{CONT}(C_k^{(i)}) = \text{CONT}(C_k^{(j)})$. ■

D. Restoring a revision

We define now the opposite function to *UPLOAD* as follows:

Definition IV.12 (Download of a component) We set

$$\begin{aligned} \text{DOWNLOAD}: \mathcal{R} &\rightarrow \mathcal{R}, \\ C_k^{(i)} &\mapsto \text{DOWNLOAD}(C_k^{(i)}) := C_k^{(j)} \\ &\text{if } (\text{UPLOAD}(C_k^{(j)}) = \text{yes} \\ &\text{and } \text{CONT}(C_k^{(j)}) = \text{CONT}(C_k^{(i)})). \end{aligned}$$

Remark IV.9 $\text{DOWNLOAD}(C_k^{(i)}) = C_k^{(j)}$ means especially that the component C_k was uploaded at the revision R_j . ■

We define the restore function having the opposite functionality to the backup function. The main difference to the usual restore strategy is that restoring the components backed up

at revision R_i is not enough, since usually only a subset of the components are backed up at a specific revision. To circumvent this impediment, the revisions at which those components have been physically uploaded are identified and are restored from those locations. When the restore operation is completed, then, the complete set of components necessary for a simulation is available.

Formally as a function:

Definition IV.13 (Restore of a revision) We set

$$\begin{aligned} \text{RESTORE}: \mathcal{R} &\rightarrow \mathcal{P}(\mathcal{R}), \\ R_i &\mapsto \text{RESTORE}(R_i) := \\ &\{C_k^{(j)} \mid k \in \mathcal{C}, \text{ such that } \text{DOWNLOAD}(C_k^{(i)}) = C_k^{(j)}\}. \end{aligned}$$

Remark IV.10 This means especially that the latest version of the components are restored. ■

We are now able to formulate the Lemma regarding the uniqueness of the upload, i.e., a new state of a component C_k during the revision process is backed up only once.

Lemma IV.2 (Uniqueness of the upload) Let $k \in \mathcal{C}$ and $v \in \text{HASH}$ be arbitrarily chosen. If $\exists j \in \mathcal{R} : \text{CONT}(C_k^{(j)}) = v$ then there exists a unique $i \in \mathcal{R}$, such that $\text{UPLOAD}(C_k^{(i)}) = \text{yes}$ and $\text{CONT}(C_k^{(i)}) = v$.

Hint Set $i := \min\{l \in \mathcal{R} \mid \text{CONT}(C_k^{(l)}) = v\}$. Then according to the definition of the status of a component (see Definition IV.8) $\text{STATUS}(C_k^{(i)}) = \text{modified}$ holds true. The result follows from the definition of the upload of the components (see Definition IV.9). ■

We can formulate now the main Lemma, which states that we have no spurious downloads.

Lemma IV.3 (Accuracy and completeness) Let $k \in \mathcal{C}$ be arbitrarily chosen. We have:

$$\begin{aligned} a) \forall i, j \in \mathcal{R} : \text{DOWNLOAD}(C_k^{(i)}) &= C_k^{(j)} \\ &\Rightarrow (\text{UPLOAD}(C_k^{(j)}) = \text{yes} \\ &\text{and } \text{CONT}(C_k^{(j)}) = \text{CONT}(C_k^{(i)})), \\ b) \forall i \in \mathcal{R} \exists j \in \mathcal{R} : (j \leq i), \\ &\text{such that } C_k^{(j)} \in \text{DOWNLOAD}(C_k^{(i)}). \end{aligned}$$

Remark IV.11 The property in a) means that we have no false downloads, i.e., each downloaded component has been uploaded some time ago. It follows from the definition of the download of a component (see Definition IV.12). The property in b) means especially that the download is complete, i.e., the latest version of each component is downloaded. It is a direct consequence of the uniqueness of the upload (see Lemma IV.2). ■

Corollary IV.2 (Complementarity) The two functions *RESTORE* and *BACKUP* are complementary, i.e.,

$$\begin{aligned} \forall k \in \mathcal{C}, \forall i \in \mathcal{R} : (C_k^{(i)} \in \text{RESTORE}(R_i)) \\ \Leftrightarrow (\exists j \in \mathcal{R} : C_k^{(j)} \in \text{BACKUP}(R_j) \\ \text{and } \text{CONT}(C_k^{(i)}) = \text{CONT}(C_k^{(j)})). \end{aligned}$$

```

1 <OpenGeoSysProject>
2   <mesh>cube_1x1x1_hex_1e0.vtu</mesh>
3   <geometry>cube_1x1x1.gml</geometry>
4
5   <processes>
6     <process>
7       <name>GW23</name>
8       <type>GROUNDWATER_FLOW</type>
9       <process_variable>pressure</process_variable>
10      <hydraulic_conductivity>K</hydraulic_conductivity>
11    </process>
12    <linear_solver>
13      <lis>-i cg -p jacobi -tol 1e-16 -maxiter 10000
14      </lis>
15      <eigen>
16        <solver_type>CG</solver_type>
17        <precon_type>jacobi</precon_type>
18        <max_iteration_step>10000</
19          max_iteration_step>
20        <error_tolerance>1e-16</error_tolerance>

```

Figure 10. Excerpt of example project file cube_1e0_neumann.prj

V. IMPLEMENTATION AND APPLICATION TO SPECIFIC USE CASE

We developed and tested AGEDRE (Automatic **G**eneration of **D**ocumentation using **R**evision control) as a prototype at UFZ and validated our theoretical concepts. We used a client / server environment at the ZIH of the Technische Universität Dresden, implementing our client in Java from scratch. On the server side, we used KIT DM [14] as the repository.

AGEDRE is a command line utility, offering the basic functionality required for a revision control system and a sophisticated error handling to deal with the complexity of the selective backup strategy on the client side and of KIT DM on the server side. In order to persist the data, AGEDRE offers two primitives, *FullUpload* as a primitive for a complete upload of all data related to a project and *Upload* as a selective backup strategy to store only modified files. Accordingly, the opposite primitives to retrieve the data are *Download*, *DownloadRevision* and *DownloadFile*. The primitive *Download* is only formally the counterpart of *Upload*, it retrieves the latest version of each file, which has been uploaded, i.e., the files of the latest revision. As mentioned, the user needs the latest version of all parameter and metadata files in order to be able to perform the simulation. In contrast, the primitive *DownloadRevision* is used to continue the modeling process from an older revision, it restores the files into the working directory, thus overwriting the latest revision. The latest revision is backed up to the file system, in order to avoid loss of data in case of inadvertent use of this primitive. The primitive *DownloadFile* has been introduced in order to restore the latest backed up version of a file, if it has been accidentally deleted or has been corrupted.

The project file (see Figure 10 for an example) is the leading file regarding the configuration of a simulation. It contains the names of the additional parameter files (namely *cube_1x1x1_hex_1e0.vtu* and *cube_1x1x1.gml*) and the configuration parameters for the simulation. Hence, each project file corresponds to a model and accordingly, the development of the model comprises modifications of the project file and the corresponding parameter files.

When the primitive *Upload* is called for a project file for

```

1   <points>
2     <point id="0" x="0" y="0" z="0"/>
3     <point id="1" x="0" y="0" z="1"/>
4     <point id="2" x="0" y="1" z="1"/>
5     <point id="3" x="0" y="1" z="0"/>
6   </points>
7
8   <surfaces>
9     <surface id="0" name="left">
10      <element p1="0" p2="1" p3="2"/>
11      <element p1="0" p2="3" p3="2"/>
12    </surface>
13    <surface id="1" name="right">
14      <element p1="4" p2="6" p3="5"/>
15      <element p1="4" p2="6" p3="7"/>
16    </surface>

```

Figure 11. Excerpt of example geometry file cube_1x1x1.gml

the first time, the corresponding dynamic flow configuration file is initialized. For an example of a flow configuration file, see the excerpts in Figures 7–9. The revision number is set to one and the cryptographic MD5 and SHA-1 hashes of each file are calculated and stored in the dynamic flow configuration file. While SHA-1 is practically collision free and it is also used by Git for integrity purposes [26], alternatively, SHA-512 could be used for enhanced security [27] [28]. For subsequent uses of the *Upload*-primitive, the cryptographic values of the parameter and metadata files are compared to the respective values stored – for previous revisions – in the flow configuration file. If the cryptographic values of file *F* is different of all the previous cryptographic values of file *F*, then the content of *F* is considered modified and it is backed up within the current revision. Otherwise, *F* is not part of the current revision. A corresponding entry is made in the dynamic flow configuration file regarding the revision under which the file – having the given content – has been backed up. Hence, the file *cube_1x1x1.gml* has the cryptographic values stored at revision 1 (see Figure 9). If the file *cube_1x1x1.gml* has, for example, not been altered at revision 2 then no similar entry is performed in the dynamic flow configuration file.

When starting the primitive *Download* – i.e., downloading the files corresponding to the latest revision – then the relevant information regarding the physical storage place of the latest version of each file – i.e., *<StorageDigitalObjectID>* – is retrieved from the dynamic flow configuration file, by considering the latest entry for each file (see Figure 9). Hence, all the related files can be accessed on the repository and retrieved accordingly.

The use case at UFZ is not designed for concurrent use. In a software development environment, users of version control systems are confronted with merge conflicts and their resolution. In contrast, simultaneous work is in the scope of our application strictly related to the model development process. The model is developed iteratively, small changes in a few parameter files can have tremendous impact on the model. Hence, members of a team can download the latest revision and can work simultaneously improving the next step. They can compare the changes of parameter files and the simulation results. Conflicts can theoretically occur but are much more unlikely due to a smaller number of users editing files and each user usually having a specific task to solve. Also, a (semi-)automatic handling of conflicts is near impossible in this scenario since

fixing conflicts requires a contextual understanding of what a specific change means in the context of a given model. Members of the team have to agree on the best outcome for the next step before uploading it as the next revision. Alternatively, they can agree on abandoning the current revision by continuing the model development from an older revision. All team members have to download the revision they agreed upon, and continue development from that point.

In addition to the dynamic flow configuration file – upon whose content they do not have direct influence – users can define and set up their own metadata files. These metadata files can contain additional – high-level or aggregated – information regarding the model development and can be used for additional documentation or for identifying model or revision characteristics. The metadata files are also very important to enable the differentiated security policy at UFZ, such that users can access the metadata files – for example by using ElasticSearch [18] – if they have the appropriate rights on the file system. In contrast, users can access simulation data (parameter files) according to their rights on the repository system KIT DM. Thus, metadata for projects is accessible for all members within the UFZ, while sensible data can only be accessed by a small number of researchers related to the project. All files of the examples can be found at [29].

VI. CONCLUSION AND FUTURE WORK

Irrespective of the fact that creating documentation is a very challenging task and that writing documentation is considered by most of the developers as an extra-effort rather than a commendation, it is rather impossible providing precise, exact, accurate, uniform and consistent documentation for developers and users requirements. Therefore, formalized automated documentation methods are necessary to develop.

The main advantage of the automatic generation of the documentation of the modeling process is the accuracy of the documentation, since there is no discrepancy between the actual generation of the model (developer's perspective) and the corresponding documentation (user's perspective). This way, we have circumvented the dilemma of writing exact manual documentation and have contributed to the paradigm change towards design and implementation of automatic documentation assuring accuracy and exactness.

Since our formal model is independent of the use case at UFZ, our approach to automatically generate a documentation for the evolution of the model during model development has a generic character and it can be applied to all domains where numerical models are developed. Moreover, the formal model is generalizable beyond the use case presented in this paper.

We have based our solution at UFZ exclusively on common techniques, which are not dependent on specific file formats or specific applications. Examples include the utilization of the XML format for the documentation, hash code based file comparison techniques as well as methods which are independent of the syntax or semantic of the underlying data. With one exception, i.e., the specification of the list of file types which should be monitored, we use only assumptions specific to the open source project OpenGeoSys [11], to applications in the earth modeling domain or to numeric simulations. The

system could version and document images, texts, or any other data in the same way.

We have opted for a revision control system using the KIT DM framework based on MASi among others, to have a completely decoupled access to the information surrounding the model development, i.e., metadata easily accessible from various locations and the model development itself, accessible only to the members of the developing team. Alternatively, using modern methods for concurrency control applied to modern database systems is a viable and future-oriented approach. This way, the difficulties on joint and concurrent development could be diminished.

To summarize, we have described in detail to what extent our versioning system facilitates simplifies and makes fully transparent the activity of application scientist. We have presented the main challenges in Section I-A, then the challenges are substantiated and concrete solutions are provided to them. Moreover, in Section IV we have set up a formal model in which the given solution is validated. This gives us hints regarding the generalization spectrum beyond the use case at UFZ.

Currently, there are no convenience features for users employing the framework. In the current implementation prototype, the executable is started in the command line, also the flow configuration file, which contains the documentation for tracking the evolution of the model is in XML format. Accordingly, additional user tests are necessary to define and implement a corresponding GUI to assure the expected readability of the document by extracting and visualizing the appropriate information. In order to build meaningful user interfaces, an intense dialog between developers and users is essential [30].

Additionally, further research is necessary to generate a high level form documentation of the changes in the parameter files. For example, when running stochastic simulations (e.g., using the Monte Carlo approach [31]) and parameters are simultaneously changed in many places, then an appropriate mechanism should be set up to assure the consistency of the changes and the creation of correct documentation.

We believe that by studying the automatically generated documentation regarding the development of the modeling workflows – especially those steps, which did not lead to a successful completion of the simulation – there is an increased possibility of knowledge extraction (by using machine learning strategies or similar techniques), such that the generation of the modeling workflows can be dramatically improved and the number of modeling steps can be considerably reduced.

ACKNOWLEDGMENT

This work was supported in parts by the German Federal Ministry of Education and Research (BMBF, 01IS14014A-D) by funding the competence center for Big Data “ScaDS Dresden/Leipzig”. This work was also supported in parts by the German Research Foundation (DFG) via the MASi project (NA 711/9-1, STO 397/4-1). We are also thankful to Dr. Nico Hoffmann (Technische Universität Dresden) for his valuable advices and comments during the developing and writing process and Dr. Agnes Sachse (né Gräbe) for her data on her hydrogeological case study.

REFERENCES

- [1] M. Zinner et al., "Automatic documentation of the development of numerical models for scientific applications using specific revision control," in ICSEA 2017, The Twelfth International Conference on Software Engineering Advances, L. Lavazza, R. Oberhauser, R. Koci, and S. Clyde, Eds., Oct. 2017, pp. 18–27, IARIA Conference. [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=icsea_2017_1_30_10110
- [2] Intergovernmental Panel on Climate Change, Climate Change 2014 – Impacts, Adaptation and Vulnerability: Regional Aspects. Cambridge University Press, 2014.
- [3] C. J. Vörösmarty et al., "Global threats to human water security and river biodiversity," *Nature*, vol. 467, 2010, pp. 555–561.
- [4] J. Grundmann, N. Schütze, G. H. Schmitz, and S. Al-Shaqsi, "Towards an integrated arid zone water management using simulation-based optimisation," *Environ Earth Sci*, vol. 65, no. 5, 2012, pp. 1381–1394.
- [5] H. Hötzel, P. Möller, and E. Rosenthal, *The Water of the Jordan Valley*. Springer, 2009.
- [6] M. Walther, J.-O. Delfs, J. Grundmann, O. Kolditz, and R. Liedl, "Saltwater intrusion modeling: Verification and application to an agricultural coastal arid region in Oman," *Journal of Computational and Applied Mathematics*, vol. 236, no. 18, 2012, pp. 4798–4809, FEMTEC 2011: 3rd International Conference on Computational Methods in Engineering and Science, May 9–13, 2011. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0377042712000659>
- [7] C. Liu, Q. Wang, C. Zou, Y. Hayashi, and T. Yasunari, "Recent trends in nitrogen flows with urbanization in the shanghai megacity and the effects on the water environment," *Environmental Science and Pollution Research*, vol. 22, no. 5, Mar 2015, pp. 3431–3440. [Online]. Available: <https://doi.org/10.1007/s11356-014-3825-4>
- [8] A. Gräbe et al., "Numerical analysis of the groundwater regime in the western Dead Sea Escarpment, Israel + West Bank," *Environ Earth Sci*, vol. 69, no. 2, 2013, pp. 571–585.
- [9] K. Rink, L. Bilke, and O. Kolditz, "Visualisation Strategies for Environmental Modelling Data," *Env Earth Sci*, vol. 72, no. 10, 2014, pp. 3857–3868.
- [10] T. Fischer, D. Naumov, S. Sattler, O. Kolditz, and M. Walther, "GO2OGS 1.0: a versatile workflow to integrate complex geological information with fault data into numerical simulation models," *Geoscientific Model Development*, vol. 8, 2015, pp. 3681–3694. [Online]. Available: <http://www.geosci-model-dev.net/8/3681/2015/>
- [11] O. Kolditz et al., "OpenGeoSys: An open source initiative for numerical simulation of thermo-hydro-mechanical/chemical (THM/C) processes in porous media," *Environ Earth Sci*, vol. 67, no. 2, 2012, pp. 589–599.
- [12] K. Rink et al., "Virtual geographic environments for water pollution control," *Int J Dig Earth*, vol. 11, no. 4, 2018, pp. 397–407.
- [13] Helmholtz Centre for Environmental Research – UFZ, "Homepage of Helmholtz Centre for Environmental Research," <https://www.ufz.de/index.php?en=34216>, retrieved: November 2018.
- [14] Karlsruhe Institute of Technology – KIT, "KIT Data Manager," <http://datamanager.kit.edu/index.php/kat-data-manager>, retrieved: November 2018.
- [15] R. Grunzke et al., "The MASi repository service - comprehensive, metadata-driven and multi-community research data management," *Future Generation Computer Systems*, 2018. [Online]. Available: <https://doi.org/10.1016/j.future.2017.12.023>
- [16] S. Chacon and B. Straub, *Git and Other Systems*. Berkeley, CA: Apress, 2014, pp. 307–356. [Online]. Available: http://dx.doi.org/10.1007/978-1-4842-0076-6_9
- [17] C. M. Pilato, B. Collins-Sussman, and B. W. Fitzpatrick, *Version control with subversion - the standard in open source version control*. O'Reilly, 2008, retrieved: November 2018. [Online]. Available: <http://www.oreilly.de/catalog/9780596510336/index.html>
- [18] C. Gormley and Z. Tong, *Elasticsearch: The Definitive Guide*, 1st ed. O'Reilly Media, Inc., 2015.
- [19] W. F. Tichy, "Rcs – a system for version control," *Software: Practice and Experience*, vol. 15, no. 7, 1985, pp. 637–654. [Online]. Available: <http://dx.doi.org/10.1002/spe.4380150703>
- [20] A. Löh, W. Swierstra, and D. Leijen, "A Principled Approach to Version Control," <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.108.8649>, retrieved: November 2018.
- [21] E. Sink, *Version Control by Example*, 1st ed. PYOW Sports Marketing, 2011.
- [22] C. L. Paris, *Automatic documentation generation: Including examples*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1995, pp. 12–25. [Online]. Available: <http://dx.doi.org/10.1007/BFb0034794>
- [23] R. Swan and J. Allan, "Automatic generation of overview timelines," in *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '00. New York, NY, USA: ACM, 2000, pp. 49–56. [Online]. Available: <http://doi.acm.org/10.1145/345508.345546>
- [24] K. McKeown, K. Kukich, and J. Shaw, "Practical issues in automatic documentation generation," in *Proceedings of the Fourth Conference on Applied Natural Language Processing*, ser. ANLC '94. Stroudsburg, PA, USA: Association for Computational Linguistics, 1994, pp. 7–14. [Online]. Available: <http://dx.doi.org/10.3115/974358.974361>
- [25] B. Möller, O. Greß, and S. Posch, "Knowing what happened - automatic documentation of image analysis processes," in *Computer Vision Systems - 8th International Conference, ICVS 2011, Sophia Antipolis, France, September 20-22, 2011. Proceedings*, 2011, pp. 1–10. [Online]. Available: https://doi.org/10.1007/978-3-642-23968-7_1
- [26] M. Stevens, E. Bursztein, P. Karpman, A. Albertini, and Y. Markov, "The first collision for full sha-1," *Cryptology ePrint Archive*, Report 2017/190, 2017, <http://eprint.iacr.org/2017/190>.
- [27] C. Dobraunig, M. Eichlseder, and F. Mendel, *Analysis of SHA-512/224 and SHA-512/256*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2015, pp. 612–630. [Online]. Available: https://doi.org/10.1007/978-3-662-48800-3_25
- [28] M. Szydio and Y. L. Yin, *Collision-Resistant Usage of MD5 and SHA-1 Via Message Preprocessing*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2006, pp. 99–114. [Online]. Available: https://doi.org/10.1007/11605805_7
- [29] D. Y. Naumov et al., "ufz/ogs-data: Initial zenodo release," Aug. 2017. [Online]. Available: <https://doi.org/10.5281/zenodo.840660>
- [30] C. Helbig, L. Bilke, H.-S. Bauer, M. Böttinger, and O. Kolditz, "Meva - an interactive visualization application for validation of multifaceted meteorological data with multiple 3d devices," *PLOS ONE*, vol. 10, no. 4, 04 2015, pp. 1–24. [Online]. Available: <https://doi.org/10.1371/journal.pone.0123811>
- [31] E. Jang et al., "Identifying the influential aquifer heterogeneity factor on nitrate reduction processes by numerical simulation," *Advances in Water Resources*, vol. 99, Jan. 2017, pp. 38–52.

Versatile but Precise Semantics for Logic-Labelled Finite State Machines

Callum McCall

Vladimir Estivill-Castro

René Hexel

School of Information and Communication Technology
Griffith University, Nathan QLD 4111, Australia
callum.mccoll@griffithuni.edu.au
v.estivill-castro@griffith.edu.au
r.hexel@griffith.edu.au

Abstract—Logic-Labelled Finite State Machines (LLFSMs) offer model-driven software development (MDSD) while enabling correctness at a high level due to their well-defined semantics that enables testing as well as formal verification. While this combination of the three elements (MDSD, validation, and verification) results in more reliable behaviour of software components, semantics is severely constrained in several areas. Here, we offer a framework that allows flexibility in execution semantics to suit specific domains while maintaining rigour and the capability to generate Kripke structures for formal verification or to execute corresponding monitoring or testing LLFSMs for validation in a test-driven development framework. Through the use of modern constructs that extend the object-oriented paradigm, the framework is able to define a set of semantics that enables versatile approaches to LLFSM definition and execution, as well as enabling functional programming constructs. This vastly increases the versatility and usefulness of LLFSMs, making them more adaptable to different domains, without sacrificing the benefits of executable models and the ability to perform formal verification.

Keywords—Logic-labelled finite-state machines; Model-Driven Engineering; Real-Time Systems; Verification; Validation.

I. INTRODUCTION

We argue here that it should be possible to rapidly and efficiently configure the semantics and Logic-Labelled Finite State Machines (LLFSMs) constructs to provide developers with the freedom to adapt or tailor the system semantics to their particular scenario. This paper shows that we can enable such versatility. We provide the capacity to instantiate new scheduling semantics with incarnations of template methods and classes, while retaining the capacity to generate corresponding Kripke structures for formal verification. The generated Kripke structures can be formally verified with standard tools [1], such as NuSMV.

By following a limited, precise semantics that is based on a synchronous concurrency model, LLFSM enable the design of software that can achieve high levels of complexity and sophistication while guaranteeing deterministic execution and facilitating formal verification [2]. The LLFSM semantics specifies precisely when variables (affected by sensors outside the system) are inspected as well as the particular points in the execution of the software where snapshots of the environment variables are taken [3]. However, this constrains the semantics of the executable model to one specific frequency and pace, which limits

the expressiveness of the designer in a way that may not be well-suited for a specific robotic or embedded target system.

Therefore, our new framework removes the need to adhere to the strict semantics currently implemented in tools such as `clfsm` [3]. Importantly, we demonstrate that maintain the ability to perform formal verification. To this end, we illustrate two areas, where we create abstractions to the LLFSM semantics and show how instantiation of these abstractions into concrete derivations maintain the ability to perform formal verification. We extend `swiftfsm` [4], a framework for LLFSMs written in `Swift`, enabling formal verification, while allowing developers more freedom to design, adapt and create new LLFSM models that suit particular, application-specific use cases.

II. LOGIC-LABELLED FINITE STATE MACHINES

Finite state machines are ubiquitous models of system behaviour. Variants of finite-state machines appear in many system modelling languages, most prominently SysML [5] and UML [6], [7], [8]. Despite their widespread use and penetration in model-driven software development, the semantics of SysML [5] and UML [9] are ambiguous [10] and restricted versions are offered to create executable models [11], real-time systems [12] or enable formal verification [13], [14]. Moreover, languages such as SysML and UML have historically adopted the event-driven form of finite-state machines inspired by Harel's STATEMATE. Unfortunately, event-driven systems cannot offer a simple semantics, as the possibility of concurrent event arrivals (e.g., from the environment or other components), can create unintended complexities emerging from subsystem interaction. This issue is intrinsic to these types of machines, where a system is modelled as being in a finite set S of states, and where transitions 'immediately' fire upon arrival of an event (more complexity usually results as event handling can itself fire a series of resulting events). The mathematical model of instantaneous (zero-time) transition is rapidly discarded because "*the software actually consumes time when processing those events*" [8, Page 50].

An example of the event-driven approach is the Specification and Description Language (SDL) formalised by the ITU-T [15]. SDL allows the modelling of reactive systems¹, and with SDL-RT [16], an extension to SDL,

¹Later we highlight the distinction between event-driven systems, reactive systems, and real-time system

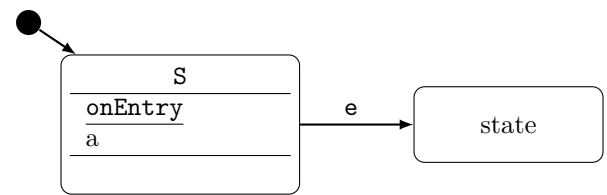
the notation aims at modelling of real-time systems. SDL (and SDL-RT) model a system as a set of asynchronously executing finite state machines called processes. Each process can communicate through the use of signals which constitute communication channels between processes. A sender does not wait for an acknowledgement from the receiver and the receiver places the signal on a queue where it remains until it is able to handle the event.

Each process may be listening for external events, i.e., events that are generated from other processes or the environment. Lamport [17] demonstrated the fundamental composability limitations of event-driven systems. Event-driven systems are designed for the best or the average case, but can result in unbounded delays or message queue sizes in the worst case, resulting in devastating consequences for real-time systems. This is because the execution of the process is completely dependent on the ordering and timing of events that the process has no control over. In extreme situations, such as an event shower, this leads to race conditions resulting in non-deterministic behaviour. To avoid running out of memory, event queues used to store events often are of a fixed size. During an event shower, such a queue can reach capacity, thus resulting in events being missed. This can therefore lead to ambiguous and nuanced behaviour, that is difficult to reproduce, as the process could potentially be in a state which it is waiting for events that have now gone undetected.

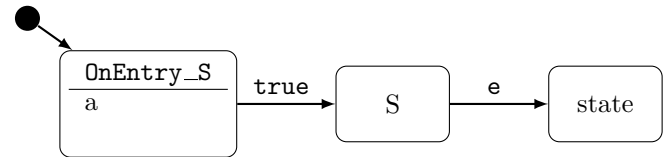
An event shower also has the added problem of making the time it takes to react to an event non-deterministic. The reason for this is that (under the standard *run-till completion* semantics [6], [8]) the system must process all events in the queue before reacting to future events. When a new event happens, the current queue size can hold a few or a large number of disparate events, making it all but impossible to predict how long the new event will be in the queue until it can be dealt with. Most notations, such as UML, SysML, and even SDL, compound this problem by allowing processes to generate new events as part of the transition actions: “A transition performs a sequence of actions. During a transition, the data of an agent is possibly manipulated and signals possibly output (depending on the content of the transition).” [18, p. 53]. That is, the firing of a transition triggers further events “instantaneously” and “simultaneously”; again, violating the ideal mathematical model that “*as close as they might be in time, events are never simultaneous*” [8, Page 50].

Therefore, an event shower, rather than been avoided by the notation, can, in fact, be fostered by the modelling language and caused by the composition of, individually benign subsystems; thus, placing a huge burden on system designers as they must now anticipate subsystem interactions, including how all possible combinations of events could influence the execution of the system. It quickly becomes impossible to discern how the combinations of all executing process contribute to the execution and timing of a particular sub-system. The solution is to move a way from the event-triggered paradigm entirely.

Complementary to event-driven fsm's, LLFSMs model a system as being in a finite set S of states. As before, each state ($s \in S$) represents a possible situation that the system may find itself in. But here it is more explicit that, while in that state, the LLFSM will execute some actions.



(a) Diagram for **Entry Actions**.



(b) Diagram providing semantics for **Entry Actions**.

Figure 1. Equivalence Wagner et al. [19] **Entry Actions** in terms of states without sections and transitions.

The system also moves from state to state by means of transitions. However, in sharp contrast with the event-driven approach, each transition is predicated by a logical expression (those familiar with UML would find this as restricting the labels of transitions to only using guards; and such models have been named *procedural* [8]). States are executable states. A state machine is not waiting for events to happen and reacting to them. Instead, it keeps executing its current state s_c , and at a precise point in the execution within its own sphere of control, the expressions labelling the associated transitions are evaluated. If one of these expressions evaluates to true, the system moves to the target state of the transition, updating the current state. Each LLFSM has a state designated as the initial state ($s_0 \in S$), representing the state at the point when execution commences.

Each state contains a set of executable actions. These actions are executed at specific times and under certain conditions. For example Wagner *et al.* define four distinct types of actions [19]:

- 1) **Entry Actions:** Executed when the system first enters a state.
- 2) **Exit Actions:** Executed when the system leaves the state.
- 3) **Transition Actions:** Executed when the system is transitioning between states.
- 4) **Input Actions:** Executed when an input satisfies a particular condition. These actions can be independent of the state.

We use Wagner *et al.* to illustrate the first point of why a general framework is of interest. We suggest that the fundamental execution cycle is the very simple notion of two states between a transition: a source state s_s and target state s_t . The distinction of an Entry Action a is merely semantic sugar for the removal of an extra state. We illustrate this in Figure 1. Wagner *et al.*'s **Entry Actions** [19] are essentially a pre-state to the state s . Figure 1a is the construct that actually has the semantics of Figure 1b. This is important, because if the expression e in Figure 1 is also **true**, it becomes very clear that the action a will be performed at least once, even if execution exits state s immediately (we note that ambiguities of this

type were already identified in standards such as SCXML).

The proper specification of semantics becomes even more important when the actions in a state access a set of variables that affect subsequent actions and transitions. That is, the attached Boolean expressions (as we mentioned, in UML and OMT these are named *guards*) involve variables. The first issue is the scope of the variables and the second issue is that of potential race conditions that could be generated due to these variables being shared in some way. Common cases of variables that are shared are the variables where sensors record a status of the environment. Thus, while the software is executing, the value of a sensor variable may change, without the software being able to control or influence the timing of these changes. Similarly, control variables for effectors are shared. The software modelled by LLFSMs may set a control variable and the driver of the effector reads such a variable to act. The `clfsm` [3] implementation of LLFSMs provides three levels of scope for variables.

- 1) **External Variables:** Variables external to the system from the perspective of the software, usually corresponding to the sensors and effectors. They may change at any point in time.
- 2) **FSM Local Variables:** These are variables that are shared between all states within a single LLFSM.
- 3) **State Local Variables:** These are variables that are local to a state.

Naturally, one can specify more variants. For example, why not have variables that are shared between all the LLFSMs of a system, but not sensors and effectors? Why not have variables whose scope is even more local than that of a state, e.g., only local to the **OnEntry** section? These examples illustrate the need for a flexible approach to extending the possibilities of LLFSM constructs. This need inspires the framework proposed in the present paper.

III. PROTOCOL ORIENTED DESIGN

Protocols in Swift are a refinement of an object-oriented construct that enables a developer to define the intent or semantics of a type without necessarily providing a concrete implementation. In this respect, protocols are similar to Java interfaces; a type that implements (or conforms to) a protocol enters into a contract, where the conforming type must implement the constraints imposed by the protocol. The advantage is that any type that conforms to the protocol conforms to the same set of semantics as any other type that also conforms to the same protocol. This enables the use of either type interchangeably. In this sense, protocols are a way to enforce semantic constraints. They enable the modelling of individual pieces of the software without the burden of defining how each piece is implemented. This creates loose coupling between types as a type may depend on a protocol (or rather the semantics which are defined within the protocol) and any type which conforms to the protocol may provide the necessary implementation.

Unlike interfaces, however, Swift takes this idea further with protocol extensions, providing a mechanism of aspect-oriented programming. This enables conforming types to receive default implementations. In other words, a conforming type may receive some or all of its implementation

for free, based on the contracts provided by protocols alone. This is not to be confused with standard class inheritance. Standard class inheritance imposes a strict implementation hierarchy and close coupling on inherited types. An implementation within a class may depend on private members or other strict types, an implementation received by a protocol extension simply models how variables and functions in the protocol can be leveraged to provide a default implementation. Importantly, a protocol extension depends only on the contract it and other protocols provide, and only on the public interface provided by these protocols.

However, the true power of a protocol extension is with conditional extensions. A conditional extension enables the developer to define rules on which conforming types receive which default implementation (if any). This way, the protocol extension can also be restrictive so that only types that conform to a specific set of protocols may receive the implementation.

Consider the example in Figure 2, showing the **Collection** protocol that contains a generic parameter **Element**. In Swift, a generic parameter on a protocol is known as an *associated type*. This protocol models a collection that contains zero or more elements. **Element** represents the type of the elements within the collection. Two concrete implementations conform to this protocol: **Array** and **Set**. These behave differently: **Set** only contains unique elements, while **Array** may contain duplicates. Swift models equality operations in the **Equatable** protocol. This protocol defines that conforming types must provide `==` and `!=` functions. A protocol extension can be created on the **Collection** protocol which provides a default implementation for the `==` and `!=` functions. These functions compare all the elements within two collections to find if they are equal, assuming that the individual elements within the collections can be compared. This is shown in Extension 1.

Extension 1. *Extend Collection where Element is Equatable:*

```
function == (lhs: Collection, rhs: Collection)
    for le in lhs, re in rhs
        if le != re
            return false
        end
    end
    return true
end

function != (lhs: Collection, rhs: Collection)
    return !(lhs == rhs)
end
```

Examples:

```
nums := [1, 2, 3]
nums == nums // Ok

people := [Person(), Person(), Person()]
people == people // Person is not Equatable
```

Now all types that conform to `Collection` (`Array`, `Set`) are able to use equality operators, assuming that the elements within the collections are able to be compared.

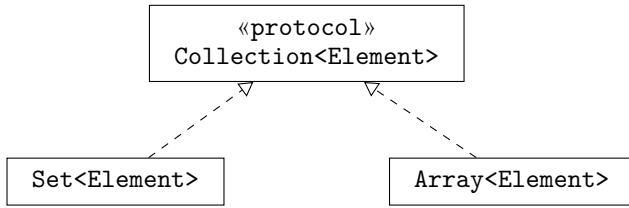


Figure 2. Collections Hierarchy

We use protocols and protocol extensions extensively in `swiftfsm` to define rules on how certain things must be implemented. This way, the developer is constrained to semantics that enable verification and model checking without otherwise restricting the language.

The `swiftfsm` framework minimises the possibility of creating a semantics whose tools for composition result in combinatorial explosion. For example, nesting in UML's statecharts is what enables the modelling of larger systems; however "*the Cartesian product machine is used as the interlingua semantics of statecharts*" [8, Page 63]. On the other hand, we shall not restrict expressivity so that only trivial systems can be modelled. We want to remain Turing complete although all properties of some executable models would not be verifiable (such as the famous Halting problem). The challenge is to enable developers to tailor the semantics to their most effective constructs while retaining small Kripke structures verifiable by standard tools.

IV. MODELLING STATES AND TRANSITIONS

We are now ready to present our first abstraction: the type used for transitions. To introduce the idea, consider the following scenario, where allowing developers to create custom semantics leads to more robust designs. Let's focus on a state *A* (Fig. 3). The `clfs`m semantics [2] explicitly specifies that the *onEntry* action will execute once and only once for each state, after which the sequence of transitions will be evaluated in the order *a*, then *b*. If the associated expression (not shown) evaluates to `true`, the corresponding transition will fire and the state will execute its *onExit* action. If none of the transitions fire, the *Internal* action will be run. In either case, the execution token passes to the next LLFSM in the arrangement.

Importantly, with this restricted semantics, it is not possible to implement an *atLeastOnce* semantics for the *Internal* action without adding another state. If transitions *a* or *b* cause a state transition, (in the `clfs`m semantics [2]), then the *Internal* action will never execute. If this functionality is required, a pattern similar to Figure 4 needs to be implemented. Note that this involves creating two states and copying (duplicating) implementation, obstructing factorisation and creating the danger of introducing failures. Both *a1* and *a3* need to be copied into the new state *A0* in order to implement the *atLeastOnce* semantics. State *A1* is almost the same as the original state *A*. This becomes arduous to maintain and modify as the

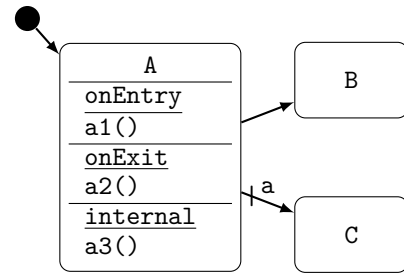
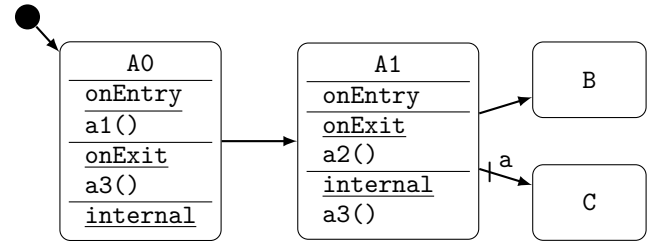
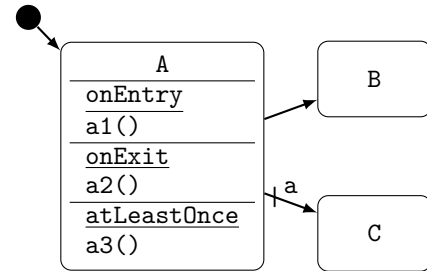


Figure 3. A simple scenario

Figure 4. Implementing "atLeastOnce" semantics in `clfs`mFigure 5. Implementing "atLeastOnce" semantics in `swiftfsm`

developer must keep the *A0* actions in sync with the *A1* actions.

With `swiftfsm`, we overcome this problem by allowing developers to define custom state types that represent such a semantics. The result is shown in Figure 5. As can be seen, the figure demonstrates how a developer may define the new *atLeastOnce* action. While this approach may seem unnecessary to apply in such a simple case, it does show the added expressiveness of the semantics which the developer may choose to employ.

Nothing is stopping the developer from creating further changes to the semantics of how states execute. Ideas such as transition actions (actions which are triggered when specific transitions are taken) become possible. The ability to create more complicated models which could utilise ideas such as state clustering or hierarchies of state machines is also viable now that a developer has the ability to proactively define the state models that they are working with. This allows the developer to tailor the tool-set to the problem they are trying to solve. Most importantly we can achieve all of this, while maintaining a strong type hierarchy and verifiable semantics as shown in Section VII.

The design of custom state types is achieved using a protocol hierarchy. Figure 6 demonstrates how this protocol hierarchy can be used to define the `clfs`m se-

mantics. First, we present the protocol at the top of the hierarchy: **Identifiable**. Instances of types conforming **Identifiable** are able to be distinguished from each other. This is achieved using the name field. Every unique instance is required to have a unique name. The **Identifiable** protocol conforms to the **Equatable** protocol, allowing equality checks by comparing the names of the two instances.

The next protocols are **StateType** and **Transitionable**. These are the protocols that are responsible for representing states and transitions. Because of the wide breadth of state models, **swiftsm** only assumes that a state has a unique name. therefore, **swiftsm** defines the **StateType** protocol only containing that name (which is inherited from **Identifiable**). The developer has complete freedom to define any number of phase actions that make up a state. This is done by defining another protocol. For the **clfsm** semantics, this is encapsulated within the **CLFSMActions** protocol. Each action that a state may perform is represented as a member function of the protocol.

The **swiftsm** framework does not even assume that a state can transition. This is a separate requirement, modelled as a separate protocol. The **Transitionable** protocol adds a sequence of transitions to conforming states. All transitions contain

- 1) a predicate function that, when it evaluates to **true**, represents a situation where the LLFSM will transition; and
- 2) a *target* state the LLFSM will transition to.

The type of the transition predicate function is defined as:

$$\text{StateContext} \rightarrow \text{Boolean}$$

This abstracts a state context type that encapsulates all (and only) the necessary variables that influence the evaluation of the predicate function. In this way, a transition function can access the necessary variables through its source state. This is an important concept when generating the corresponding Kripke structure of an executable model in order to perform formal verification. We profit and explicitly use referential transparency for the generation of Kripke structures. That is, transitions emanating from a state will be evaluated with a fixed state context variation that has no further dependencies or side-effects.

This allows for an important optimisation. Typically, an LLFSM state corresponds to several Kripke states, because of

- state sections (e.g., **onEntry**, **onExit**, **Internal**, **atLeastOnce**, etc.), and
- the potential semantics of snapshotting external variables between these state sections.

However, our semantics recognises that external variables that are not involved in a transition will not need to create a new transition evaluation context. Therefore, the above transition type is side-effect free and removes the need to consider all possible combinations of external variables outside those appearing in the transaction.

The traditional conceptualisation of the class of transitions is that transitions have a source and a target state. Such a conceptualisation complicates the optimisation we

just mentioned, as the transition is in a static relationship with its source state (typically implemented as a reference). Our approach does not need to change the source state of a transition in an LLFSM to create the Kripke states for sections. Our framework only updates the possible changes to the external variables of relevance, and submits the State with this new context for evaluation to the transition. Importantly, this means that the evaluation of any transition is referentially transparent, as it is a pure function with explicit inputs and outputs and no side-effects. The Kripke structure generated in this way is guaranteed to obtain the effect of evaluation of the transition without possible side effects influencing the transition as all the variables are in the context attached to the state.

Notice that **CLFSMState** does not contain an **addTransition** function. This is provided by a default implementation shown in Extension 2

Extension 2. *Extend StateType where Self is Transitionable:*

```
function addTransition(t: _TransitionType ...)
    transitions := transitions ∪ t
end
```

V. MODELLING LOGIC-LABELLED FINITE STATE MACHINES

The ability to create custom LLFSM semantics can be beneficial. Within the confines of the **clfsm** semantics, an LLFSM is capable of being suspended, restarted or stopped. The LLFSM is also responsible for executing states. While this semantics is expressive, it can be somewhat restrictive. For example, why not make it possible to allow an LLFSM to control a given set of LLFSMs in a master/slave relationship? This would make it possible to create strict LLFSM hierarchies which form a tree structure. This is quite different from the LLFSM hierarchies described in the literature [20] that does not enforce any hierarchy and allows any LLFSM to control any other LLFSM. Our stricter approach here helps ensure clarity and safety.

The way in which **swiftsm** models LLFSMs allows the developer to define custom semantics is again achieved using a protocol hierarchy (Figure 7). The LLFSM protocol hierarchy, while more complex, follows the same methods we employed to create the state semantics hierarchy. A base protocol is used and further features are added using separate protocols.

We first present the simplest protocol **StateContainer**. This protocol contains no members, instead it simply defines the **_StateType** associated type. This associated type must be defined to allow conforming types to refer to a single **StateType** type. That is to say, types that are conforming to **StateContainer** are manipulating or interacting with a single type of state in some way.

The first protocol which conforms to **StateContainer** is **FiniteStateMachineType**. This protocol represents an LLFSM. However, similar to how the **StateType** protocol made no assumptions about the actions contained within a state, the **FiniteStateMachineType** protocol makes

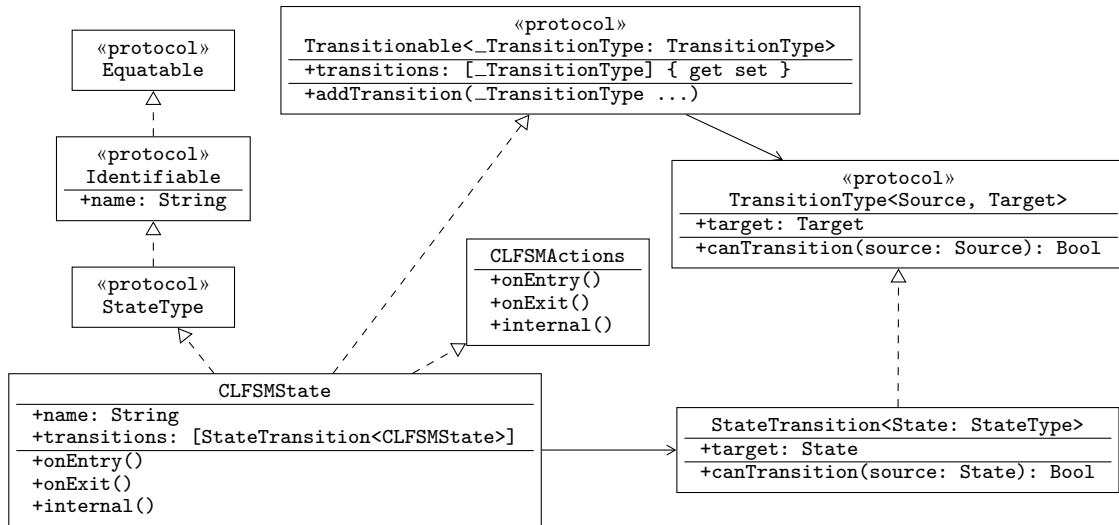


Figure 6. Defining The cl fsm State Semantics

no assumptions on what the LLFSM model may do. The `FiniteStateMachineType` protocol does not even assume that it will execute a state. This is modelled by the `StateExecutor` protocol.

We now introduce a new abstraction over the original concept of an LLFSM ringlet [2]. A ringlet defines how the sections within a state are executed, and more specifically, how and in what order each action is executed. We propose to view ringlets as pure functions that take a state and return the next state to execute. Therefore, we have them as objects of the following type.

$$\text{State} \rightarrow \text{State}.$$

If a new state is returned, then the LLFSM has transitioned. By modelling a ringlet in this fashion, we enable developers to create custom ringlets that determine how their states are executed. To illustrate the adaptability of this approach, it is also possible to create different ringlets that execute the same set of states in different ways. Importantly, the execution of the state becomes orthogonal to the definition of the state.

However, in practice it is common that a ringlet may require to modify state information. To this end, the `swiftfsm` framework provides the `Ringlet` protocol, which defines an `execute` function. If we look at previous semantics for LLFSMs, and in particular to the semantics offered by the `cl fsm` compiler, we can see that the ringlet only executes the `onEntry` section when the previously executed state does not equal the state currently being executed (in particular, if a state has a transition to itself, this is a legal construct, but if the transaction executes, in `cl fsm` this does not re-run the `onEntry` section). If a developer wishes to extend the semantics that all arriving transitions (including self-transitions) cause the `onEntry` section to execute, our framework here allows the creation of a `CLFSMRinglet` that contains a `previousState` member variable that the `execute` function refers to and manages when executing the current state. That is we are using the `Method` pattern, and the developer supplies the method that defines the specific ringlet to sequence sections of a

state.

The `StateExecutor` protocol (in combination with `IncrementalExecutor`) provides a `next` function. This function is responsible for executing the current state. However, if the LLFSM contains a ringlet, then the `next` function may delegate the execution of the state to the ringlet. This functionality is encapsulated in the `StateExecutorDelegator` protocol that defines a `StateExecutor`, which contains a `Ringlet`. `FiniteStateMachineType` provides a default implementation for the `next` function when conforming to `StateExecutorDelegator`. This is shown in Extension 3.

A similar pattern is followed for providing the remaining features of the LLFSM. In this sense, a feature is modelled as a protocol, and further protocols are created to enable default implementations. A further example is with the `Restartable` protocol. This protocol provides a `restart` function which, when called, should restart the LLFSM so that the initial state becomes the current state.

Extension 3. *Extend `FiniteStateMachineType` where `Self` is `StateExecutorDelegator` and `Self` is `PreviousStateContainer` and `__StateType = RingletType.__StateType`:*

```

function next()
    temp := ringlet.execute(currentState)
    previousState := currentState
    currentState := temp
end
  
```

However, recall that the `cl fsm` semantics state that the `onEntry` action is only executed if the previous state does not equal the current state. If the LLFSM was truly to restart, then the previous state should also reset to its initial value. This way, there does not exist a situation where the LLFSM does not execute the `onEntry` action. To provide the default implementation

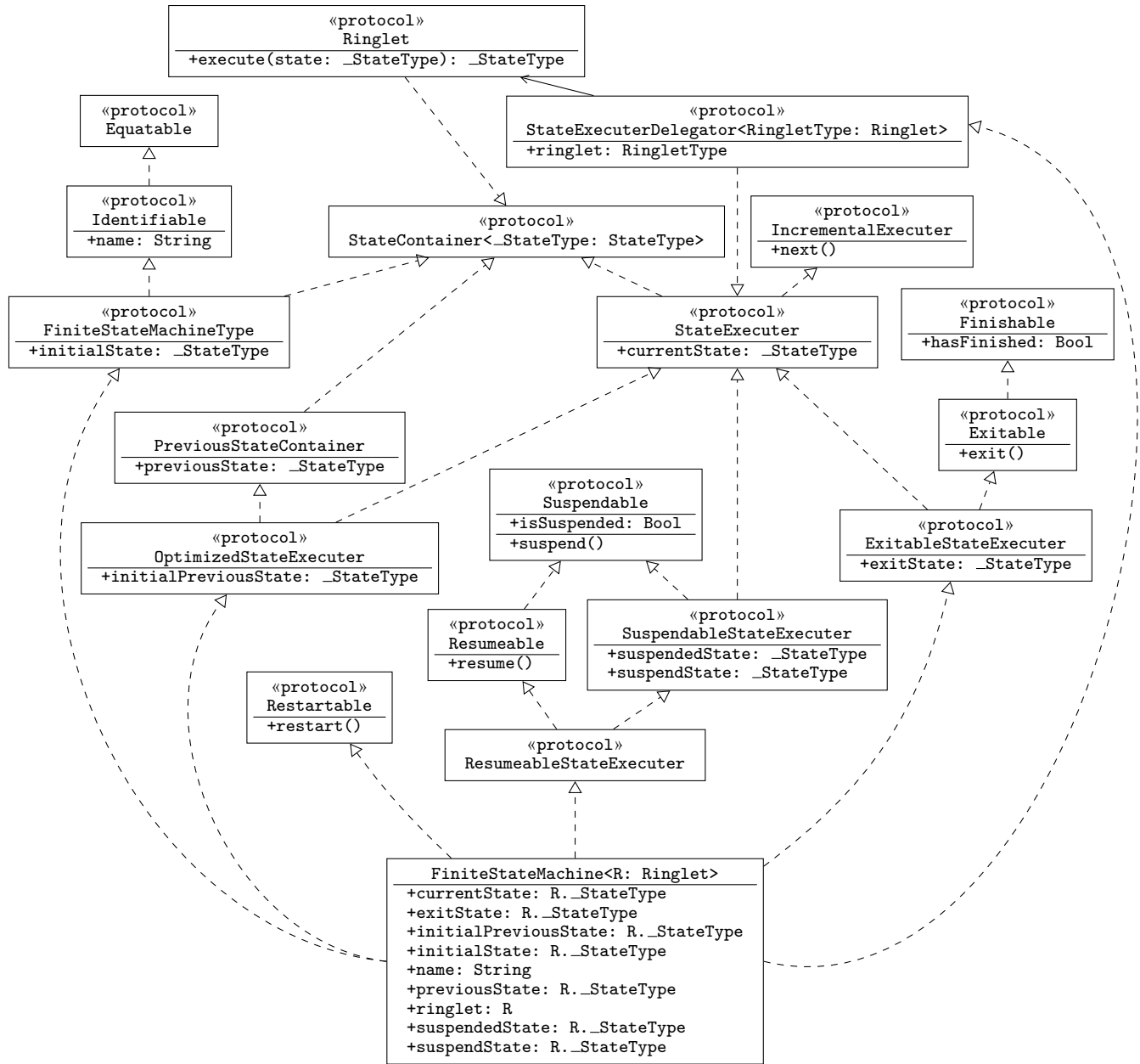


Figure 7. The LLFSM Protocol Hierarchy

for a restart function, the `FiniteStateMachineType` protocol requires that there should also exist a conformance to `OptimizedStateExecutor`. This protocol enables an `initialPreviousState` member which represents the first previous state. With this, a default implementation for the `restart` function is now possible. The `restart` function effectively sets the current state and the previous state to their initial values.

To put everything together, the `FiniteStateMachine` type implements the default model in `swiftfsm`. This implements the same semantics as `clfsm`. Notice, however, that the `FiniteStateMachine` type does not need to provide any function implementations, as the type receives its entire implementation from protocol extensions.

VI. SCHEDULING

Because LLFSM are not event-driven, they are scheduled using a round robin scheduler. We provide such scheduling as the default in the framework `swiftfsm`. Therefore, a single ringlet, for the current state of each LLFSM, is executed in a sequential fashion. This creates concurrent execution in a predictable manner reducing state explosion for formal verification. The sequential execution avoids thread management and avoids complexities associated with parallel execution, (there are essentially no critical sections or mutual exclusion challenges). Because of the sequential scheduling, we have a deterministic execution of an arrangement of the LLFSMs, thus when the Kripke structure is created for the entire arrangement, we have a

smaller Kripke model (a smaller NuSMV input file) that with unconstrained concurrency of event-driven systems. By preventing side-effects (as shown in the previous Section), we further reduce the size of the Kripke structure enhancing the feasibility of performing model checking.

Furthermore, **swiftfsm** uses a stricter snapshot semantics when executing the ringlets. A snapshot is taken of the external variables before the ringlet is executed. The state then uses the snapshot when executing actions and evaluating transitions (recall our execution context). Only once the ringlet has finished executing, any modifications appear visible externally (e.g., to the environment). This defines the granularity at which the system is reactive to changes observable by sensors in the environment and does not need to make a dangerous assumption of well-behaved environments and that the software always runs faster than any external part of the system. Compare this with many formal verification approaches that only work with ideal event-driven systems, that do not exist in practice. We detach from such ideal conceptions that “events consume no time: they are zero time episodes” [8, Page 50] and “as close as they might be in time, events are never simultaneous” [8, Page 50] because even in UML it is extremely easy to create an event for which there would be several listeners whose response would be the generation of an event (creating simultaneous events). Or admitting upon further analysis that “the software actually consumes time when processing those events” [8, Page 50]. Thus, we avoid approaches where extended finite-state machines handling of external variables is simply assumed to be irrelevant: “During a macrostep, the values of the inputs do not change and no new external events may arrive; in other words, the system is assumed to be infinitely faster than the environment” [21, p. 172]. Alternatively, the environment is assumed to be well-behaved, so that it sends the input the software requires at the right time, forming “a closed model corresponding to the complete mathematical simulation of the pair formed by the software controller and the environment” [22, p. 89]. We also do not follow the simplistic approach that assumes any external stimulus (change of external variables) will not happen until all internal changes take place “giving priority to internal actions over external actions” [23].

We argue that the specification of when a snapshot is taken defines the level of atomicity of the sections within the state run by the ringlet with respect to the external variables. This becomes particularly important when performing formal verification.

VII. FORMAL VERIFICATION

If one strictly follows the derivation of Kripke structures from the artefact of sequential program constructs [24], the corresponding Kripke states would not only be the boundaries of sections of LLFSM states, but every assignment and operation in those sections correspond to extended FSMs, containing programming language statements (e.g., in **Swift**). The sequential execution of LLFSMs and its default snapshot semantics enables more succinct Kripke structures, where the delicate point is the handling of the external variables [25], [26]. Nevertheless, as we mentioned, such a default semantics requires recording all of the variables influencing the execution before and

after every state section in order to generate the Kripke structure [25]. For consistency, we configured a version of **swiftfsm** that followed such an approach [4].

These earlier approaches relied on the ringlet itself to record variables, influencing the execution of a state. However, a more succinct approach can be used and a further optimisation can be made. Since the **swiftfsm** framework not only uses a sequential scheduling similar to **clfsm**, but a ringlet’s execution is atomic with respect to the external variables, ringlet execution can now be treated as a black box.

Consequently, a snapshot should only be taken of the variables before and after the entire ringlet for a state’s execution. This variation also prevents statements being executed that make modification to variables that are not reflected in the final context for the next Kripke state. For example, a state may make changes to an external variable during an **onEntry** section that is cancelled by a further modification in the **onExit** section. Since no effect of this will occur during the state’s execution, as we now identify a Kripke state *before* and *after* an entire ringlet execution, interim changes are not reflected in the resulting Kripke structure.

Importantly, we argue that this is a benefit, not a problem! In **swiftfsm**, the statements within sections of the state operate within a context derived from a snapshot of the external variables, which gets taken precisely when the state is scheduled. There is absolutely no way that any modification could (nor should) affect the environment until the snapshot is saved. External variables are updated precisely once when the ringlet has finished executing. Similarly, since **swiftfsm** uses sequential scheduling, there is no way for the modification of non-external variables to have side-effects and influence the execution of other machines, because the semantics is equivalent to a single thread. The only important record for the construction of the Kripke states (to be part of the Kripke structure or verification) is the context (of the variables) before and after each ringlet is executed.

To demonstrate the approach for creating a Kripke structure, consider the **Incrementing LLFSM** shown in Figure 8. This LLFSM is responsible for incrementing a counter by one. When the value of the counter reaches a non-zero number, the LLFSM transitions to the **Exit** accepting state. However, this LLFSM provides two boolean external variables, each representing the state of a button in the environment. When both buttons are pushed, the LLFSM transitions to the **Exit** state regardless of the value of the counter.

Using the round-robin sequential scheduler requires that a snapshot be taken before and after a state is executed. This is represented in the Kripke structure. An execution of a state results in several nodes. Firstly, taking a snapshot will result in a node being created for each combination of external variables. For the **Incrementing LLFSM**, this will result in a total of 4 nodes. However, the advantage with **swiftfsm** is, as stated previously, treating the execution of a ringlet as an atomic action. Each 4 nodes that represent the reading of a snapshot will each transition to a single node. We therefore classify each node within the Kripke structure as an *R* (read) node or a *W* (write) node. *R* nodes represent the state of the system

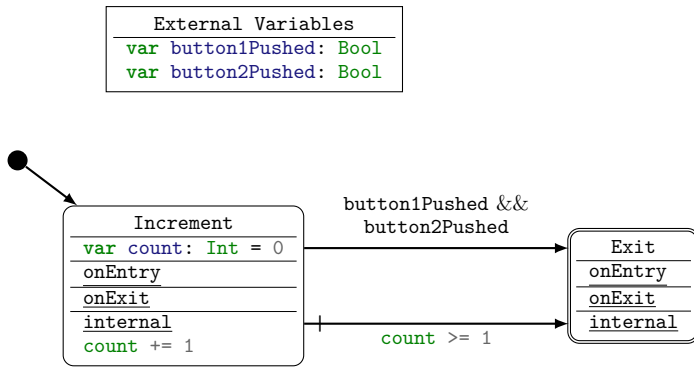


Figure 8. The Incrementing LLFSM

before a ringlet has been executed, whereas the W nodes represent the state of the system after the ringlet has finished executing. When executing using the round-robin scheduler, a single ringlet execution will result in multiple R nodes each leading to a single W node. This is shown in Figure 9.

VIII. MICROWAVE CASE STUDY

We present a case study where we simplify the model of a microwave oven, a ubiquitous example in the software engineering literature of behaviour modelling through states and transitions [27]. This model has been extensively studied in formal verification [24, p. 39], as the safety feature of *disable cooking when the door is open* is analogous to the requirement that a radiation machine should have a halt-sensor [28, p. 2]. Software models for microwave behaviour are widely discussed [29], [30], [31], [32], [33], [34]). Figure 10 shows the standard executable model with LLFSMs. While this model is transparent and formal verification establishes requirements, the full machinery of Kripke states for each of the three state-sections is not required (note that all **Internal** sections are empty and the only **onExit** section that is used is in the timer LLFSM, in state 3 to `Add_1_Minute`). Moreover, the model would also be simplified if the `timeLeft` variable were to be removed by making it equivalent to the condition `0 < currentTime`. With respect to the requirements specified in Myers and Dromey [34, p. 27, Table 1] or in Shlaer and Mellor [30, p. 36] the behaviour of such a simplification is irrelevant. But, for model checking, removing the Boolean variable `timeLeft` alone would half the number of Kripke states (and the corresponding size of the NuSMV file where formal verification is conducted is thus halved). By removing the state sections, the number of Kripke states would be halved again. Thus, it would be advantageous to derive LLFSMs, where states have no **onExit** nor **Internal** actions.

The new model would globally replace `timeLeft` by `0 < currentTime`. All declarations of `extern timeLeft` disappear from all LLFSMs. Thus, the timer machine changes to Figure 11. This change results in slightly different behaviour. With the executable model of Figure 10, when the button is pressed for the first time and not released, nothing would happen. Now, when the button is pressed for the first time and not released, if the door is closed, cooking will commence and the light will go on. As long

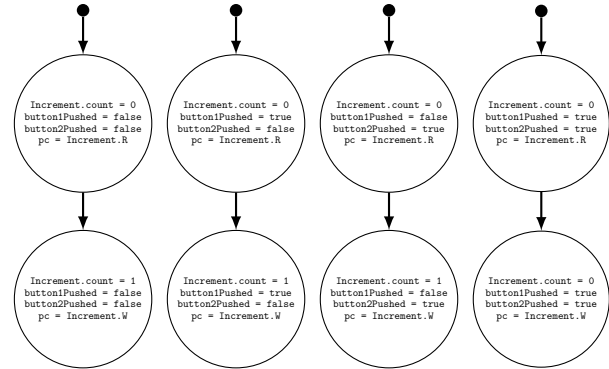


Figure 9. Executing the initial ringlet of the Incrementing LLFSM

as the button is pressed and not released such cooking with the light on will continue and the timer will not be decremented. This behaviour does exist in a slightly similar form in Figure 10, but only happens from the second time onwards. That is, the user must press the button; upon releasing the button, cooking starts and the light turns on. If the user presses and holds the button now that cooking has started, it also blocks timing counting down. Again, we do not consider this subtle difference in behaviour relevant as it is never identified in the requirements (Figure 12 illustrates the concurrent execution of the arrangement of LLFSMs and documents their state changes for a use-case). However, the variation simplifies the Kripke structure radically for more efficient formal verification of the requirements. With our framework, the designers can easily alternate between the two executable models, and conduct model checking on both.

A further optimisation can be made when considering how `swiftfsm` currently handles the snapshots of external variables. Recall that a snapshot is taken before the ringlet executes, and then saved back to the environment once the ringlet has finished executing. By changing these semantics to a per-schedule cycle, as opposed to a per-ringlet cycle, we can further minimise the number of Kripke States that are generated. Taking the microwave as an example, instead of taking a snapshot of the external variables before executing each state, we instead take a single snapshot of the environment for each execution of one schedule over the arrangement of LLFSMs. Each LLFSM would therefore share the same snapshot and any modifications made to the snapshot will only be saved once all LLFSMs have executed their current state.

This has a drastic impact to the number of Kripke States that are generated for the Kripke Structure. Consider all possible combinations of a snapshot of the external variables. The microwave uses three Boolean variables, therefore this results in $2^3 = 8$ possible combinations. There are normally four snapshots taken per schedule cycle as there are four LLFSMs executing and a snapshot is taken when a ringlet in each LLFSM is executed. Therefore, there are $2^{3 \times 4} = 4096$ possible combinations of snapshots per schedule cycle. When taking a single snapshot at the start of the schedule cycle, the result is $2^3 = 8$ possible combinations of snapshots. Removing the `timeLeft` variable further reduces this to $2^2 = 4$ combinations of

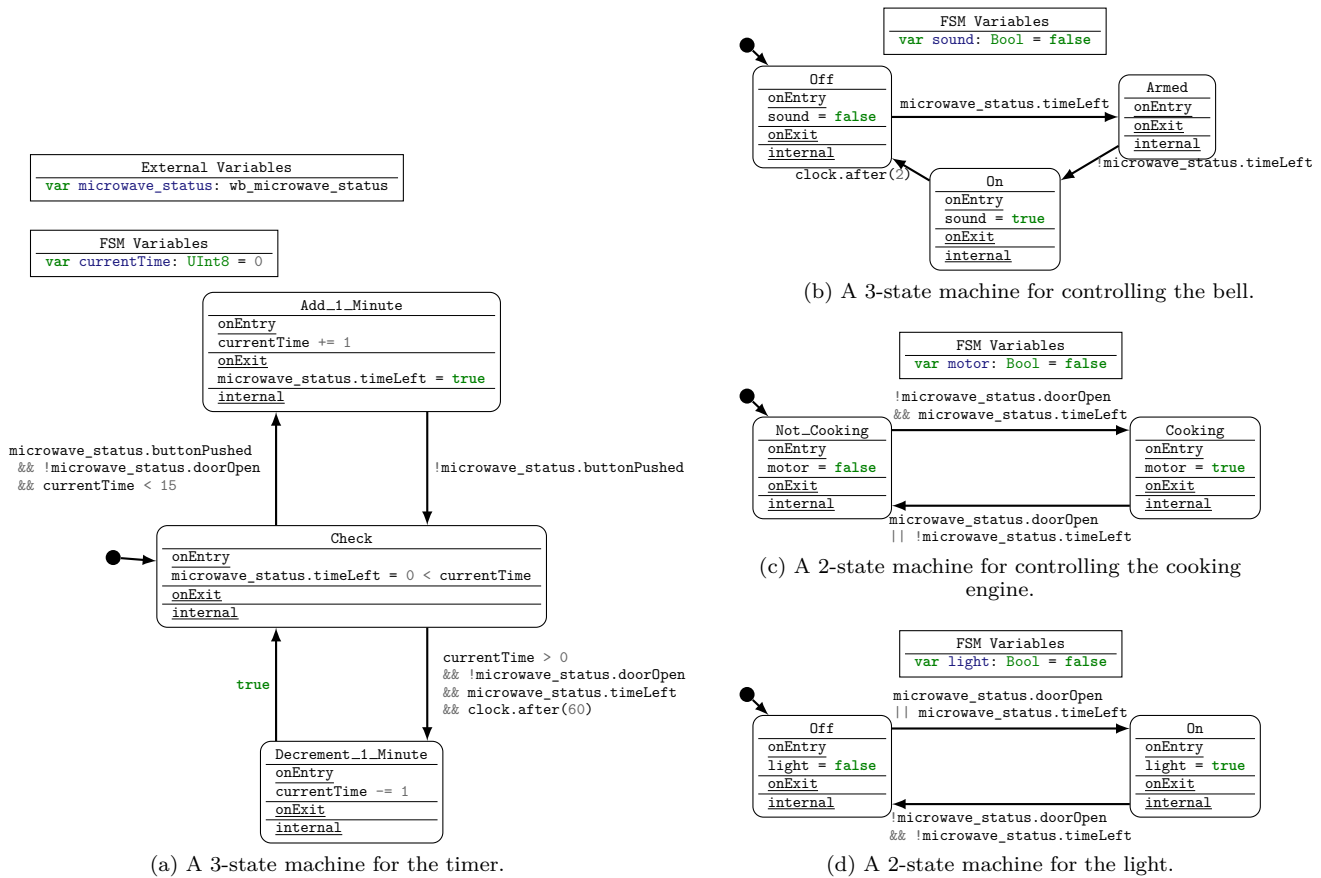
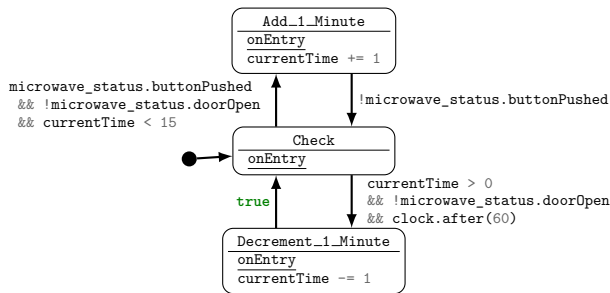


Figure 10. Complete model of one-minute microwave.

Figure 11. Simplified timer with **onEntry** sections only.

snapshots per schedule cycle, a reduction by three orders of magnitude.

To demonstrate the effect that limiting the snapshots has on the Kripke structure size, we provide a git repository which contains the Kripke structures in several formats for the **Incrementing LLFSM** and the one-minute microwave **LLFSMs**. When following the original semantics (those where a snapshot of the external variables is taken and saved before and after each ringlet execution) the Kripke structure for the microwave contains 369 260 nodes. By using a single snapshot semantics where a snapshot is taken per schedule cycle, the Kripke structure for the microwave contains 60 674 nodes. A massive reduction.

The Kripke structures can be located at <https://github.com/mipalgu/VersatileKripkeStructures>.

IX. CONCLUSION

In software engineering, there is a prevalence for modelling using UML state charts (which is a derivation of Harel's State Charts [35]) and which are event-driven. Moreover, Sommerville [27], states that "state models are often used to describe real-time systems" [27, p. 544], citing UML. We note that Sommerville also uses a microwave to illustrate how **FSMs** model the behaviour of systems [27, p. 136]. Because of these associations among systems that respond to stimuli, it is important to clarify the terminology regarding what constitutes an event-driven system, a reactive system and more importantly, a real-time system. Reactive-systems are responsive systems without much processing, as opposed to deliberative systems (which reason, plan, learn) [36].

We refer to an event-driven system as one typically based on a software architecture built around stimuli-driven call-backs, a subscribe mechanism and listeners that enact such call-backs (very much as GUIs are composed for desktops today). Reacting to stimuli in this way implies uncontrolled concurrency (e.g., using separate threads or event queues). Event-driven programming is also illustrative of this mechanism that follows the inversion of control (IoC) design principle; call-backs are custom code only

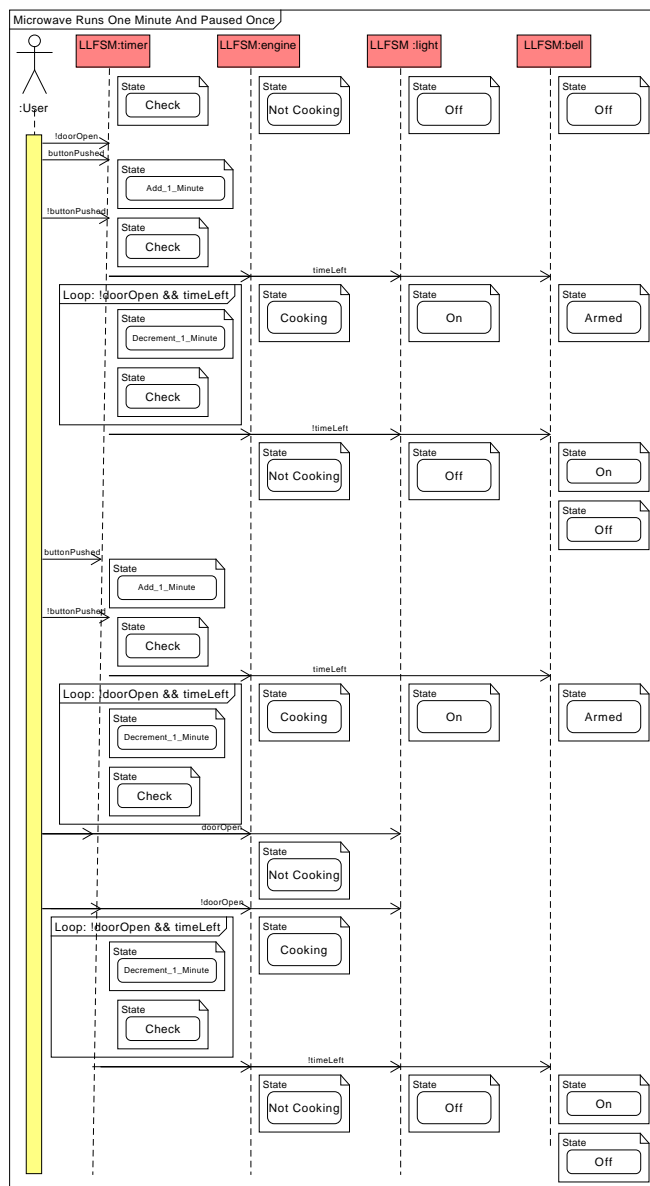


Figure 12. States transitions in a sequence diagram for the use case of running the microwave for one minute and a pause in another minute.

concerned with the handling of events, whereas the event loop and dispatch of events/messages is handled by the framework or the runtime environment. This application of the “Hollywood Principle: Don’t call us, we’ll call you”, while very productive in several contexts (we already mention GUI applications), has serious limitations for real-time applications. We insist that fundamental (mathematically supported assertions) have long been established [17] regarding the limitations of event-driven systems. The counterpart to event-driven systems are time-triggered systems.

Real-time systems are required to meet time-deadlines in response to stimuli [37]. Therefore, although closely related, these terms are not the same, and in this pa-

per, we argue (supported by the work of Lamport [17]) that there are many solid reasons why real-time systems may be better served by time-triggered systems and pre-determined schedules, rather than the unbounded delays that may occur in event-driven systems.

There is nothing wrong with using a loosely defined semantics in visual and textual notations (in particular the event-driven state charts of UML) when communicating the main ideas of software designs to stakeholders [38]. However, advocates of model-driven software development (MDS) argue that software developers shall mostly work with models and that UML is the programming language [39]. However, in this scenario the UML has shortcomings. For example, the relevance of MDS to the synthesis of behaviour intended for embedded systems has been strongly emphasised [40]; particularly stressing that UML state charts are the prevalent diagrammatic tool for behaviour. But, for instance, modelling behaviour for FPGAs has consistently avoided the event-driven approach of UML state charts: Wood et al. [40] reviewed earlier attempts to generate VHDL from UML and found that either 1) translation was incomplete (aimed at simulation and not hardware synthesis), 2) covered a tiny subset of UML’s state chart notation or 3) were far from following closely an MDS methodology. Wood et al. [40] attempt to directly define a model transformation semantics (from the syntactic construct of UML state charts to the VHDL code) also ran into issues; in particular, the asynchronous nature of UML was replaced by synchronous specification in VHDL [40, Page 1362] (other semantic changes are also listed [40, Page 1364], as well as a table of unsupported features [40, Table 11]). Wood et al. [40] removed the run-until-completion semantics of UML requiring designs where all the consequences of an event must terminate by the next clock click [40]. They eliminated events from labelling transitions, and now a Boolean expression of the form `event_has_happened` (that pool the state of an input signal) replaces each instance of an event. Thus, their UML models allow only guards that monitor signals (they enforce a system of syntactic priorities in case several signals become true in nesting state charts, but for each model, transitions must cover all cases and be mutually exclusive). Thus, we have another scenario where LLFSMs are being used with precise but particular semantics.

The work presented in this paper illustrates how LLFSMs can be used as executable models. Moreover, we argue that their deterministic execution and verifiability is more suitable for real-time systems than systems where threads proliferate. In this paper, we have introduced a flexible semantic model for logic-labelled finite-state machines. Compared to traditional event-driven state machines and LLFSMs, our approach allows redefining while retaining precision of the semantics of executable models [6], [7]. Our framework allows high-level, executable models, which are less error-prone and eliminate duplication. Moreover, we have shown these semantics can be modelled in a referentially transparent way that creates simpler Kripke structures, allowing formal verification of our executable models, that is orders of magnitudes faster for the same model than previous approaches.

REFERENCES

- [1] C. McColl, V. Estivill-Castro, and R. Hexel, "An oo and functional framework for versatile semantics of logic-labelled finite state machines," in The Twelfth International Conference on Software Engineering Advances, ICSEA 17, J. Lavazza, R. Oberhauser, R. Koci, and S. Clyde, Eds. IARIA, October 8th-12th 2017, pp. 238–243.
- [2] V. Estivill-Castro and R. Hexel, "Arrangements of finite-state machines - semantics, simulation, and model checking," in MODELSWARD, S. Hammoudi, L. F. Pires, J. Filipe, and R. C. das Neves, Eds. SciTePress, 2013, pp. 182–189.
- [3] V. Estivill-Castro, R. Hexel, and C. Lusty, "High performance relaying of C++11 objects across processes and logic-labeled finite-state machines," in Simulation, Modeling, and Programming for Autonomous Robots: 4th International Conference, SIMPAR 2014. Springer International Publishing, October 20th-23rd 2014, pp. 182–194.
- [4] C. McColl, "swiftfs - A Finite State Machines Scheduler," Honours Thesis, Griffith University, 170 Kessels Rd, Nathan QLD, 4111, Australia, 10 2016.
- [5] S. Friedenthal, A. Moore, and R. Steiner, A Practical Guide to SysML: The systems Modeling Language. San Mateo, CA: Morgan Kaufmann Publishers, 2009.
- [6] M. Samek, Practical UML Statecharts in C/C++, Second Edition: Event-Driven Programming for Embedded Systems. Newton, MA, USA: Newnes, 2008.
- [7] D. Pilone and N. Pitman, UML 2.0 in a Nutshell. O'Reilly Media, Inc., 2005.
- [8] D. Drusinsky, Modeling and Verification Using UML Statecharts: A Working Guide to Reactive System Design, Runtime Monitoring and Execution-based Model Checking. Newnes, 2006.
- [9] M. Fowler, UML Distilled: A Brief Guide to the Standard Object Modeling Language, 3rd ed. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2003.
- [10] R. Rumpe, "Executable modeling with UML – a vision or a nightmare? –," in Issues and Trends of Information Technology Management in Contemporary Associations Volume 1, M. Khosrowpour, Ed. Idea Group Publishing, May 19th-22nd 2002, pp. 697–701.
- [11] S. J. Mellor and M. Balcer, Executable UML: A foundation for model-driven architecture. Reading, MA: Addison-Wesley Publishing Co., 2002.
- [12] B. P. Douglass, Real Time UML: Advances in the UML for Real-Time Systems (3rd Edition). Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 2004.
- [13] A. Krupp, O. Lundkvist, T. Schattkowsky, and C. Snook, "The adaptive cruise controller case study — visualisation, validation, and temporal verification," in UML-B Specification for Proven Embedded Systems Design, J. Mermet, Ed. Springer US, 2004, pp. 199–210.
- [14] B. Selic and S. Grard, Modeling and Analysis of Real-Time and Embedded Systems with UML and MARTE: Developing Cyber-Physical Systems, 1st ed. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2013.
- [15] Recommendation ITU-T Z.100: Specification and Description Language – Overview of SDL-2010, International Telecommunication Union, April 2016. [Online]. Available: <https://www.itu.int/rec/T-REC-Z.100-201604-I/en>
- [16] Specification and description language - real time. [Online]. Available: <http://www.sdl-rt.org/>
- [17] L. Lamport, "Using time instead of timeout for fault-tolerant distributed systems," ACM Transactions on Programming Languages and Systems, vol. 6, 1984, pp. 254–280.
- [18] Recommendation ITU-T Z.101: Specification and Description Language – Basic SDL–2010, International Telecommunication Union, April 2016. [Online]. Available: <https://www.itu.int/rec/T-REC-Z.101-201604-I/en>
- [19] F. Wagner, R. Schmuki, T. Wagner, and P. Wolstenholme, Modeling Software with Finite State Machines: A Practical Approach. 6000 Broken Sound Parkway NW, Boca Raton, FL 33487-2742: CRC Press Taylor & Francis Group, 2006.
- [20] V. Estivill-Castro and R. Hexel, "Verifiable parameterised behaviour models for robotic and embedded systems," in International Conference on Model-Driven Engineering and Software Development, MODELSWARD 2018, vol. 1. SCITEPRESS Science and Technology Publications, January 22nd-24th 2018, pp. 364–371.
- [21] W. Chan, R. J. Anderson, P. Beame, D. Notkin, D. H. Jones, and W. E. Warner, "Optimizing symbolic model checking for statecharts," IEEE Trans. Softw. Eng., vol. 27, no. 2, Feb. 2001, pp. 170–190.
- [22] J.-R. Abrial, Modeling in Event-B - System and Software Engineering. Cambridge University Press, 2010.
- [23] L. Grunske, K. Winter, N. Yatapanage, S. Zafar, and P. A. Lindsay, "Experience with fault injection experiments for FMEA," Software, Practice and Experience, vol. 41, no. 11, 2011, pp. 1233–1258.
- [24] E. M. Clarke, O. Grumberg, and D. Peled, Model checking. MIT Press, 2001.
- [25] V. Estivill-Castro and D. A. Rosenblueth, Model Checking of Transition-Labeled Finite-State Machines. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, pp. 61–73.
- [26] V. Estivill-Castro, R. Hexel, and D. A. Rosenblueth, "Efficient modelling of embedded software systems and their formal verification," in 19th Asia-Pacific Software Engineering Conference, vol. 1, Dec 2012, pp. 428–433.
- [27] I. Sommerville, Software engineering (9th ed.). Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 2010.
- [28] C. Baier and J.-P. Katoen, Principles of model checking. MIT Press, 2008.
- [29] S. J. Mellor, "Embedded systems in UML," OMG White paper, 2007, www.omg.org/news/whitepapers/ label: "We can generate Systems Today" Retrieved: April 2017.
- [30] S. Shlaer and S. J. Mellor, Object lifecycles : modeling the world in states. Englewood Cliffs, N.J.: Yourdon Press, 1992.
- [31] F. Wagner, R. Schmuki, T. Wagner, and P. Wolstenholme, Modeling Software with Finite State Machines: A Practical Approach. NY: CRC Press, 2006.
- [32] L. Wen and R. G. Dromey, "From requirements change to design change: A formal path," in 2nd International Conference on Software Engineering and Formal Methods (SEFM 2004). Beijing, China: IEEE Computer Society, 28-30 September 2004, pp. 104–113.
- [33] R. G. Dromey and D. Powell, "Early requirements defect detection," TickIT Journal, vol. 4Q05, 2005, pp. 3–13.
- [34] T. Myers and R. G. Dromey, "From requirements to embedded software - formalising the key steps," in 20th Australian Software Engineering Conference (ASWEC). Gold Coast, Australia: IEEE Computer Society, 14-17 April 2009, pp. 23–33.
- [35] D. Harel and M. Politi, Modeling Reactive Systems with Statecharts: The Statemate Approach. New York, NY, USA: McGraw-Hill, Inc., 1998.
- [36] R. C. Arkin, Behavior-Based Robotics. Cambridge, Mass.: MIT Press, 1998.
- [37] H. Kopetz, Real-Time Systems - Design Principles for Distributed Embedded Applications, 2nd ed., ser. Real-Time Systems Series. Berlin: Springer, 2011.
- [38] W. J. Dzidek, E. Arisholm, and L. C. Briand, "A realistic empirical evaluation of the costs and benefits of UML in software maintenance," IEEE Trans. Softw. Eng., vol. 34, no. 3, May 2008, pp. 407–432.
- [39] B. Selic, "The pragmatics of model-driven development," IEEE Software, vol. 20, no. 5, Sept 2003, pp. 19–25.
- [40] S. K. Wood, D. H. Akehurst, O. Uzenkov, W. G. J. Howells, and K. D. McDonald-Maier, "A model-driven development approach to mapping UML state diagrams to synthesizable VHDL," IEEE Transactions on Computers, vol. 57, no. 10, Oct 2008, pp. 1357–1371.

Pragmatic Approach to Automated Testing of Mobile Applications with Non-Native Graphic User Interface

Maxim Mozgovoy, Evgeny Pyshkin

School of Computer Science and Engineering, Division of Information Systems

University of Aizu

Aizu-Wakamatsu, Japan

E-mail: {mozgovoy, pyshe}@u-aizu.ac.jp

Abstract—This article addresses the problem of automated smoke testing for mobile applications with hand-drawn non-native graphic user interface (GUI) within the context of continuous integration pipeline. In such applications the traditional approach to define and test situations triggered by appearance of certain GUI elements accessed programmatically does not work, so we need to apply image recognition and pattern matching algorithms to testing both the application interface and its major functional features. We introduce one example, which is a Unity-based mobile game “World of Tennis: Roaring ’20s”. Our idea is to classify GUI elements (including buttons, game control elements, static and movable objects) with respect to their appearance in different type of game scenes, as well as to find pattern recognition methods providing the best similarity values to increase GUI element recognition quality and, therefore, to suggest a reliable support for test script writers.

Keywords—*software testing; GUI; image recognition; pattern matching; similarity; mobile game; continuous integration.*

I. INTRODUCTION

This post-conference article is an extended revision of our paper presented at the UBICOMM-2017 conference [1].

Automated testing is an integral element of software development pipeline, frequently discussed in literature. Though many specialists agree that automated tests could not completely substitute careful manual testing [2], the combination of automated tests with manual quality assurance procedures is one of the central tenets of established software development methodologies, such as test-driven development [3] and behavior-driven development [4]. In addition, testing frameworks assure better communication between developers and customers: they allow developers rediscovering the customer context better and can be used to improve acceptance testing practices and procedures, which, in turn, are essential parts of iterative software development processes [5][6][7].

In practice, however, maintaining an adequate set of tests can be a challenging and time-consuming task: surveys show that the majority of professional developers are not satisfied with their current testing suites or do no automatic testing at all, complaining that the tests are difficult to write and maintain [8]. A pragmatic approach to testing suggests prioritizing testing strategies, and keeping at least the most useful tests well maintained. Some authors suggest giving

the priority to smoke tests that check basic functions of the whole software system [9]. Let us recall that, according to [10], smoke tests represent a subset of all defined/planned test cases that cover the main functionality of a component or system. Their goal is to ascertain that the most crucial functions of a program work correctly, without checking more fine-grained aspects of software’s functional specification. By definition, smoke tests do not cover the code under testing completely.

Humble and Farley believe that smoke tests (considered as elements of software deployment process) are probably the most important tests to write [11]. In turn, Mustafa et al. advise to “stick to smoke testing” in case of severe time and cost pressure [12]; MSDN documentation calls smoke testing “the most cost effective method for identifying and fixing defects in software” after code reviews [13].

Thus, smoke tests are aimed at performing some basic checkups: whether the program runs at all, is it able to open required windows, does it react properly to user input, etc. Automated user interface (UI) smoke tests should be able to access applications in the same way as users do, so they need to manipulate application’s user interface. Specifically, testing graphical UI (GUI) provides an interesting and nontrivial case of testing automation [14].

While a smoke test can be as simple as “launching the application and checking to make sure that the main screen comes up with the expected content” [11], it can also evolve into a complex suite of tests checking core application functionality. Complex testing scenarios may require the use of specialized smoke testing frameworks. One interesting and widespread example of such scenario is mobile application testing automation. Mobile apps are hard to test due to several factors:

1. All supported platforms and a wide range of devices should be used in tests;
2. The apps should be tested on real devices rather than on emulators/simulators;
3. The tests should reveal both bugs and problems such as battery drain and low performance;
4. Non-native (hand-drawn) GUI requires specialized handling.

The idea of hiding platform-specific UI automation frameworks behind a universal interface was recently implemented in the tools such as Appium [15] and Calabash [16]. However, these frameworks can only interact with user interfaces based on native GUI components of an

underlying operating system (such as widgets exposed by standardized GUI libraries like Qt or WinForms). Therefore, additional efforts are required to recognize and interact with non-native GUI elements, referenced in test scripts. For example, cross-platform mobile games often rely on such hand-drawn GUI elements. These widgets might look slightly different on different devices with different resolutions; their onscreen positions often are not fixed. Many interactions also have to be performed with “active” game objects, such as buildings, game characters, map elements, etc. Technically, an operating system “sees” non-native GUI elements as graphical primitives drawn on a canvas, and, therefore, cannot manipulate them via standard object-oriented API.

Consequently, a UI automation framework also recognizes the main window of a non-native GUI based application as a plain graphical image containing no UI elements. Similar problems might appear in other multimedia projects, such as text recognition applications (where texts are represented as images), applications based on interactive electronic maps, etc.



Figure 1. Actual screen of *World of Tennis: Roaring '20s*.

The example we use in this article is the mobile game project “World of Tennis: Roaring ’20s”, where we are involved in [17] (see Figure 1). This game is made with Unity, and its GUI is represented with hand-drawn components. This setup makes difficult to develop standard automated GUI tests and basic functional smoke tests, since all screen elements are in fact plain graphical images that we cannot easily access programmatically in test scripts [18]. Hence, testing automation requires integrated use of image recognition and pattern matching capabilities.

The basic goal of this paper is to show how standard pattern recognition tools can be used as a universal aid for GUI testing (primarily, for applications with non-native user interface). We list a number of practical challenges associated with this approach, and discuss how one can fine-tune the settings of the pattern recognition procedure to ensure smooth operation in a variety of scenarios.

The paper has the following structure. In Section II, we describe our approach within the context of existing research in the area. In Section III, we examine a number of problems to be resolved while implementing test scripts using pattern recognition methods. Section IV describes how our

experiments were organized. Section V introduces a discussion on applicability of the suggested approach for wider range of mobile applications. In Section VI, we briefly summarize the current state of this project and introduce the tasks for future work.

II. APPROACH AND RELATED WORK

In our previous work, we described the process of deployment of smoke testing infrastructure using Appium as a testing automation framework and continuous integration setup using TeamCity as a build server [14] (see Figure 2). We also demonstrated that identifying objects of interest on the screen, such as GUI elements or game characters, could not be completely reduced to the task of perfect matching of a bitmap image inside a screenshot [1]. It happens due to several reasons:

- Onscreen objects may be rendered differently with different GPUs or rendering quality settings;
- Screens vary in dimensions, so patterns might need scaling;
- Onscreen objects often intersect with each other, so one object might partially hide another object.

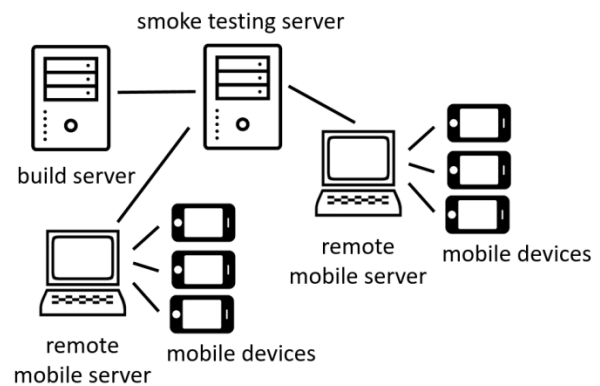


Figure 2. Mobile application testing infrastructure.

Thus, the most straightforward way to recognize such elements is to rely on approximate pattern matching. There are several tutorials where an idea of using image matching in creating test scripts is discussed [19][20]. OpenCV library [21] provides a number of methods for pattern recognition and can serve as a typical tool used for searching and finding the occurrences of the given pattern in a larger image. Basic OpenCV pattern matching methods can be accessed using `matchTemplate()` function with a parameter defining a specific method among the variety of supported pattern matching methods [22][23]:

1. CV_TM_SQDIFF: square difference matching minimizing the squared difference between the pattern and the image area;
2. CV_TM_SQDIFF_NORMED: normalized version of the square difference matching (normalized methods are typically used when the effects of lighting difference between a pattern and an image should be reduced [24]);

3. CV_TM_CCORR: correlation matching method multiplicatively matching a template against the image and then maximizing the matched area;
4. CV_TM_CCORR_NORMED: normalized version of the correlation matching method;
5. CV_TM_CCOEFF: correlation coefficient matching method that matches a template against the image relative to their means and generates a matching score ranging from -1 (complete mismatch) to 1 (perfect match); and
6. CV_TM_CCOEFF_NORMED: normalized version of the correlation coefficient matching method.

As we know from different sources (such as [24]), the *matchTemplate()* function slides a template over the given area and computes similarity value in a range of [0..1] for each pixel location, thus maximizing pattern matching similarity. The function yields the best value as the final recognition similarity, so we are able to analyze the result from the viewpoint of GUI elements recognition quality.

An automated test consists of the following steps:

- Take a game screenshot.
- Detect the presence of a certain GUI element using image recognition.
- React properly.
- Check the expected application behavior or program state.
- Repeat the process.

Hereafter, we describe the core function of the automated smoke tests we developed for the “World of Tennis: Roaring ’20s” mobile game. The Python test script presented below is responsible for checking application initialization and several actions performed in the beginning of the game. Initial game run requires several core subsystems to work properly. Thus, successful first run is more than just a smoke test; it is a good indicator of a stable game build. In general, the test script follows the same routine as described in the list above. In the current automated testing framework implementation, we match GUI elements with OpenCV *matchTemplate()* function called with a parameter TM_CCOEFF_NORMED.

Since the game may run on devices with different screen sizes, we scale the screenshots to match the dimensions of the original screen used to record graphical patterns. In Listing 1 we present a Python code for the *findByImage()* function. This function tries to find a template GUI element pattern *templateImg* in the screenshot *img* (highlighted line), and returns the similarity score we got from the corresponding OpenCV algorithm paired with the coordinates of the matched area center.

LISTING 1. FINDING A TEMPLATE WITH TM_CCOEFF_NORMED METHOD

```
import cv2 # OpenCV
import imutils

def findByImage(img, templateImg):
    img_h, img_w = img.shape[0:2] # image dimensions
    template = cv2.imread(templateImg, 1) # read template
    h, w = template.shape[0:2]

    # rescale the template for the target device's screen
    # (here we assume that template image was taken
    # at 1920x1080 resolution)
```

```
factor = float(img_w) / 1920
template = imutils.resize(template,
    width = int(w * factor), inter = cv2.INTER_CUBIC)
h, w = template.shape[0:2]

res = cv2.matchTemplate(img, template,
    cv2.TM_CCOEFF_NORMED)
(_, maxVal, _, maxLoc) = cv2.minMaxLoc(res)
result = ((maxLoc[0] + (w / 2),
    maxLoc[1] + (h / 2)), maxVal)
return result
```

The function *waitFor()* (see Listing 2) waits for the given image or a list of images to appear on the screen (while the application is running). In case of failed recognition, an exception is thrown. This function uses the above presented *findByImage()* in order to recognize the GUI element. The similarity score returned by *findByImage()* function is then compared to the globally defined threshold. After that, it is possible to interact with a GUI element.

LISTING 2. WAITING FOR A GUI ELEMENT IMAGE ON THE SCREEN

```
def waitFor(driver, tries, interval, imagefiles):
    # convert a single image into a one-element list
    if type(imagefiles) is not list:
        imagefiles = [imagefiles]

    for i in range(tries):
        img = takeScreenshot(driver) # a wrapper for Appium
        # screenshot function

        for imagefile in imagefiles:
            (loc, simRatio) = findByImage(img, imagefile)
            if simRatio > Config.PicFoundThreshold:
                return (loc, simRatio)
        time.sleep(interval)

    raise RuntimeError("GUI element not found. Tried: " +
        str(imagefiles) + ".")
```

The fragment of function *testFirstRun()* (see Listing 3) presents the first part of the smoke test responsible for checking some initial actions during the game run:

1. Test whether the user name is properly entered and accepted by the application.
2. Test whether the screen with players appears after pushing OK button.
3. Test a possibility to go to the configuration screen for the selected player.
4. Test whether the tabs work properly in the configuration window.
5. Test how the selected configuration is applied after tapping Apply button.

LISTING 3. FIRST RUN SMOKE TEST (FRAGMENT)

```
# Test first run (test script fragment)
def testFirstRun(driver):
    print("Testing the first run")

    print("wait for the name input box, and tap it")
    tap(driver, waitFor(driver, 9, 3, 'name_input_box.png'))

    print("type the user name")
    # this function generates and types a random user name
    inputName(driver)

    print("tap OK button")
    tap(driver, waitFor(driver, 3, 3, 'ok_button.png'))
    time.sleep(4)

    print("wait for the player circle, and tap it")
    tap(driver, waitFor(driver, 10, 3,
        ['player_green_circle.png', 'finger.png']))
```



```

print("tap 'Character'")
tap(driver, waitFor(driver, 3, 3,
    ['character_tab_1.png', 'character_tab_2.png']))

print("tap on the player")
tap(driver, waitFor(driver, 3, 3,
    'player_select_region.png'))

print("tap 'Apply'")
tap(driver, waitFor(driver, 3, 3,
    ['apply_button_1.png', 'apply_button_2.png']))

# ...

```

In *testFirstRun()* script we used a number of GUI elements (Figure 3) that we are trying to recognize on the screens presented in Figure 4.



Figure 3. GUI elements used in the first run test script.



Figure 4. Game screenshots used in the first run test script.

III. TOWARDS BETTER GUI ELEMENT RECOGNITION RELIABILITY

As it follows from the observations mentioned in Section II, an important problem is to find optimal parameters of image recognition algorithms maximizing GUI elements recognition reliability. Such an approach would decrease the number of automated tests that might fail, not because of the software bugs, but due to the UI elements recognition defects.

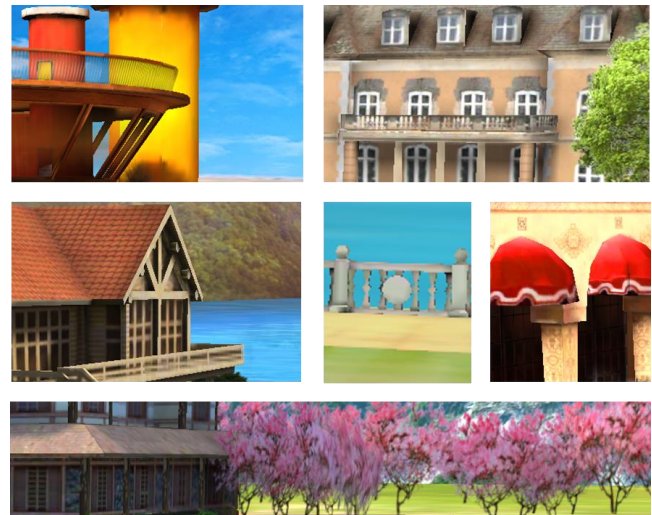


Figure 5. Club view template examples that can be used for checking screen orientation.

Both types of recognition errors (false negatives and false positives) actually caused problems in our experiments. Let us provide some examples.

- The first step in all our tests is to detect whether the device screen is rotated upside down (for the sake of brevity, this procedure was omitted in Section II). To do it, we try to match certain elements of the initial “club view” scene against a number of regular and flipped patterns (see Figure 5). Our experience shows that this step often provides false positives, as both types of patterns can be found by OpenCV algorithms.
- When the game designers slightly change the buttons (in order to beautify them, make them slightly larger or smaller, change fonts, colors, etc.), our tests stop recognizing them. It may happen even with very simple elements like buttons shown in Figure 6.
- When buttons have two states (enabled/disabled), visually shown with different colors, the tests often fail to recognize them accurately.
- Additional elements shown on or next to GUI elements (such as checkboxes or numbers) might prevent the tests to recognize them properly.

The challenge is to make sure that we still can match changing GUI elements, while being able to distinguish them.



Figure 6. Static UI controls: buttons, tabs, check boxes, static images, etc.

There are also numerous moving objects on the screen. Suppose the test script needs to press on the character’s model in the pictures shown in Figure 7.

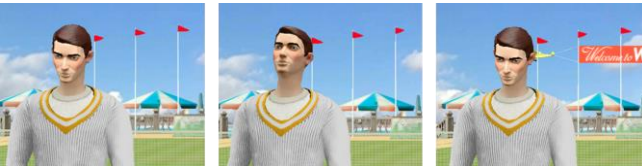


Figure 7. Moving objects: the object view changes and the surrounding area might change as well.

An animated head might make a perfect match difficult not only because of changes in object view itself, but also because of possible changes in the adjacent screen area (e.g., an airplane appeared in the sky, in our case). Hence, it might be required to work with a set of different images related to

the same UI element and to perform a matching process for all of them. We have to consider a possibility to work with a larger region providing necessary context to avoid false positive recognition results. Our hypothesis is that experimenting with different pattern matching algorithms will allow us to provide a number of recommendations for test script developers. These recommendations will provide hints on what are the better algorithms to use in certain test contexts.

IV. EXPERIMENTS WITH DIFFERENT UI ELEMENTS

A large variety of UI components designed for the “World of Tennis: Roaring ’20’s” allows us to classify them in a number of classes including the following UI types:

- UI widgets: buttons, edit boxes, tabs, etc.
- Static images: player portraits, court fragments, popup message boxes.
- Dynamic objects: moving player figures, onscreen hints.

As discussed in Section II, for the first implementation of test scripts, we used OpenCV *matchTemplate()* function and a number of built-in pattern matching methods. After experimenting with a number of test scripts, we realized that pattern matching reliability significantly depends on a recognition task. For example, simple button-like GUI elements (buttons, menus, tabs) can usually be recognized with high degree of similarity (0.90..0.98), according to OpenCV reports. Similarity score decreases to (0.63..0.65) for certain elements interfering with the background like menu items placed against the sky with moving clouds. This makes perfect template matching impossible in principle. Lower similarity values might occur even for the objects that are not graphically complex, but contain patterns distorted during rescaling (as Figure 8 shows).

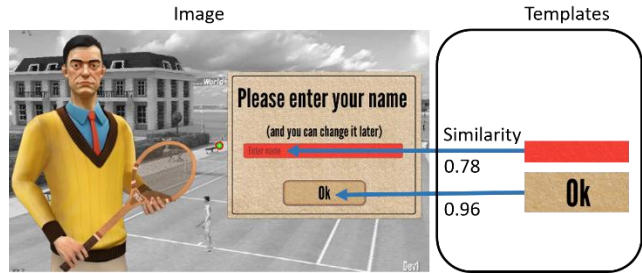


Figure 8. Template matching similarity varies for different UI elements.

Even in the simplest cases, the similarity scores might differ depending on the device where the code is running. As Table I illustrates, the scores for OK button range from 0.94 to 0.99 for different devices even if there is no any recognition complication such as “bad” background, surrounding or changing objects, etc. (apparently due to different screen resolutions and screen image scaling distortions).

Table I lists a selection of devices used in this experiment, their screen characteristics and reported similarity scores [1].

TABLE I. EXPERIMENTING WITH OK BUTTON USING TM_CCOEFF_NORMED ALGORITHM

Case	Description of the test case			
	Device	Screen	Tap size	Similarity
a	Xiaomi Redmi Note 3 Pro	1920x1080	1920x1080	0.99
b	iPad Air	2046x1536	1024x768	0.95
c	Doogee X5 Max Pro	1280x720	1280x720	0.94

In certain tests, the system reports the presence of UI elements that are actually not shown on the screen. Such false positive cases typically happen if some similar-looking graphical elements are confused with each other, especially when they are surrounded by moving objects or complex background. As noted in Section III, a possible way to struggle with such cases is to try to match larger regions in order to include more context into the pattern. For example, in our tennis game application, the Skip button is always placed next to a checkbox, so we can try to match the whole button/checkbox region.

One more approach that can be used to decrease interference with the complex background is to apply image transformations for both the grabbed screenshot and the pattern [25]. In particular, we were experimenting with a number of edge detection filters including Laplace algorithm and Canny edge detection algorithm [26]. Samples of image transformations are shown in Figure 9. We used the GNU Image Manipulation Program [27] for Laplace edge detection and the online tool “Imaging Web Demonstrations” [28] for Canny edge detection algorithm.

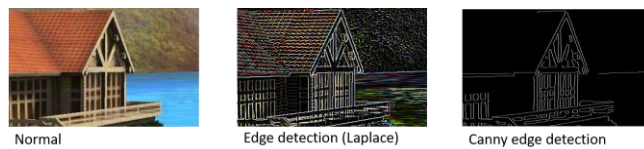


Figure 9. Samples of filtered templates.

Preliminary experiments show that the use of transformed images does not significantly improve recognition of UI elements when an element is present on the screen. However, such transformations may be helpful to avoid false positives.

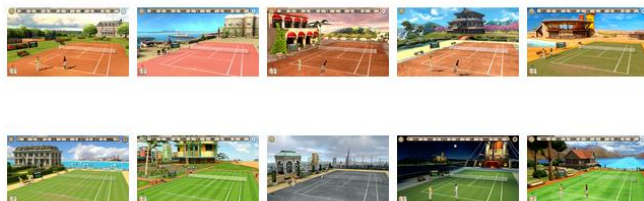


Figure 10. Scenes used for experimenting with false positive cases.

In Table II, we summarized our experiments with the TM_CCOEFF_NORMED algorithm, as described in details in our separate paper [25]. For the experiments, we used 10

game scene screenshots (Figure 10) and 11 test fragments. 10 fragments were taken from the above-mentioned screenshots (hence, each fragment exists exactly in one scene). We also used one pattern fragment that does not belong to any of 10 screenshots. Thus, from 110 possible combinations only 10 correspond to true positive cases.

As we can see from the results listed in Table II, edge filtering does not affect true positive cases. However, the number of false positive cases with substantially high scores significantly decreases if the pattern matching method receives filtered images as input.

TABLE II. RECOGNIZING A SELECTION OF SAMPLE FRAGMENTS WITH TM_CCOEFF_NORMED METHOD IN PLAIN CASE AND WITH EDGE DETECTION FILTERS

	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
True positive (normal images)	0	0	0	0	0	0	1	0	1	8
True positive (Canny)	0	0	0	0	0	0	1	0	1	0
False positive (normal images)	0	0	14	19	32	22	13	0	0	0
False positive (Canny)	58	37	5	0	0	0	0	0	0	0

Figure 11 provides visual demonstration of the fact that the scores for false positive cases are shifting to the lower ranges after applying an edge detection filter to both scene and template images.

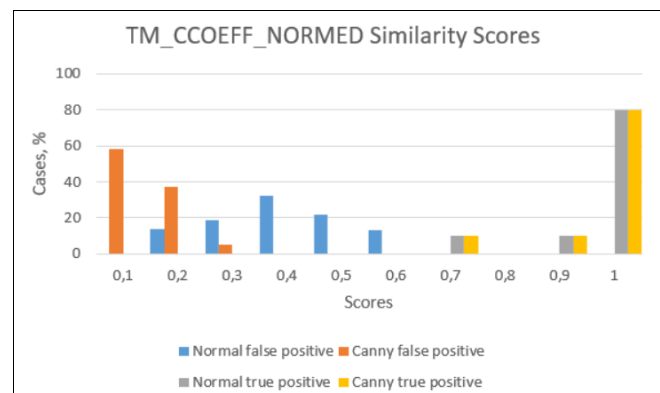


Figure 11. Struggling with false positive cases using Canny edge filtering.

In principle, there is no necessity to strive for the highest similarity scores in the situations where the GUI elements are present on the screen. Our primary goal is to minimize the number of both false positive and false negative errors: what we need is an algorithm that separates the scenes *with* and *without* target GUI elements reliably.

We have to run pattern-matching algorithms for significantly different usage contexts: in player settings window, club selection window, game selection window,

ongoing game window, etc. We expect that for every combination (*algorithm, UI element, usage context*), the reported similarity values could give us better understanding how to improve the quality of test scripts and, therefore, how to make the next steps towards building a testing automation framework for mobile applications based on hand-drawn or non-native GUI components.

V. DISCUSSION

The approach to GUI testing we describe above is applicable to a large variety of applications implementing custom (non-native) user interface. They include desktop and mobile games, HTML5 Canvas-based web applications, and many specialized instruments such as graphical or audio editors or mapping software. Accessing UI elements via the API of the underlying operating system is the preferable way for interacting with application for most tests. However, it is not available if the application UI is not based on standard widgets.

One may argue that using pattern recognition leads to fragile tests, since any changes in application GUI might break test scripts. While this is certainly the case, we have to note that all kinds of GUI tests are prone to fragility, and might fail due to reorganization, renaming, or addition of new UI elements. Writing robust GUI tests is a challenging task regardless of the approach used.

Reliance on screenshot processing slows down the testing process considerably. There are three major reasons. First, the process of screenshot generation on a mobile platform might take several seconds depending on a particular device. Second, the screenshot needs to be transferred to the machine running the test scripts, which requires fast and reliable connection. Third, pattern recognition is also computationally intensive, though in our experiments with five concurrent test processes the largest performance bottlenecks were still caused by the procedure of taking and transferring the next screenshot to the testing machine.

However, let us note that the screenshots provide a useful graphical log of the testing process, and we often resort to it for debugging purposes. So one might consider taking screenshots regardless of the method of accessing GUI controls. Furthermore, accessing UI elements with conventional methods (such as using XPath locators in Appium) can also be slow, and might take several seconds per query depending on a particular situation.

VI. CONCLUSION AND FUTURE WORK

Let us note that image recognition algorithms are rarely discussed within the scope of software testing, so we believe that advancing and improving the quality of the proposed approach will provide a feasible solution to be used as a part of integration pipeline in software development and testing.

Current frameworks such as Appium [29] allow running smoke tests on real mobile devices. However, they do not provide built-in capabilities for managing multiple-device tests and for integration of smoke testing into continuous delivery pipeline. Several companies (such as Bitbar and Amazon) offer “mobile test farm” services. They are also used and evaluated by academic researchers [30][31].

However, they are expensive for small developers, offer a limited range of mobile devices, and lack flexibility. There is also an open Smartphone Test Farm initiative [32], but its primary goal is to provide remote control options for Android devices rather than to build an automated cross-platform smoke testing facility. We also have to mention a number of commercial service providers, such as Amazon, Xamarin, and Bitbar supporting heterogeneous multiple-device farm facilities

Our further goal is to create an open source framework for small-scale mobile farms [33]. The aim of this framework is to let anyone to quickly connect their own iOS or Android devices into a fully functional mobile farm, and integrate it into existing continuous delivery pipeline. Our software will make smoke testing of mobile apps easier to set up for the developers, and, therefore, will increase the popularity of automated testing methods, and, consequently, will contribute to the improved quality of software. We believe that a practical testing framework should implement the following capabilities:

1. Concurrent tests on several mobile devices.
2. Automated detailed reports of test results with app screenshots and activity logs.
3. Health monitoring of the devices for early detection of battery drain or device malfunction.
4. Manual and event-driven test runs.

We expect that the approach can be advanced towards designing the architecture of a farm as a distributed system allowing geographically dispersed teams to use their devices effectively. We specifically address the tasks of automating apps with non-native GUIs, such as apps written in Unity (a popular instrument for cross-platform development of games and multimedia software). This project can be considered a transdisciplinary human-centric research [34] that requires applying the solutions achieved in areas such as pattern recognition, intelligent interfaces and usability to a distinct application domain (mobile software testing), rather than straightforward integrated use of available tools and methods.

REFERENCES

- [1] M. Mozgovoy and E. Pyshkin, “Using Image Recognition for Testing Hand-drawn Graphic User Interfaces,” 11th International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies (UBICOMM 2017), Barcelona, Spain, November 12-16, 2017, IARIA, pp. 25-28.
- [2] D.M. Rafi, K.R. Moses, K. Petersen, and M. Mäntylä, “Benefits and limitations of automated software testing: systematic literature review and practitioner survey,” In Proceedings of the 7th International Workshop on Automation of Software Test (AST '12), IEEE Press, Piscataway, NJ, USA, 2012, pp. 36-42.
- [3] K. Beck, Test-Driven Development by Example. Addison-Wesley Professional, 2002, 240 p.
- [4] S. Bellware, “Behavior-Driven Development,” Code Magazine, 2008, vol. 9(3).
- [5] E. Pyshkin, M. Mozgovoy, and M. Glukhikh, “On requirements for acceptance testing automation tools in behavior driven software development,” In Proceedings of the 8th Software Engineering Conference in Russia (CEE-SECR), 2012, accessed: November 13, 2018. [Online]. Available:

- http://2012.secrus.org/2012/presentations/pyshkin-mozgovoy-glukhikh_80_article.pdf.
- [6] Boehm. B. Software Engineering Economics. Prentice Hall, Englewood Cliffs, NJ, 1981.
- [7] E. Pyshkin and M. Glukhikh, "Teaching program flow validation: A case study of branch coverage testing," In Ari Lindeman (Ed.), Studies in social sciences, humanities and engineering, The second joint research publication of Peter the Great St. Petersburg Polytechnic University and Kymenlaakso University of Applied Sciences, Kymenlaakso University of Applied Sciences, Kouvola, Finland, 2015, Series A, No. 71, pp. 72-80.
- [8] E. Daka and G. Fraser, "A Survey on Unit Testing Practices and Problems," 25th IEEE International Symposium on Software Reliability Engineering (ISSRE), 2014, pp. 201-211.
- [9] S. McConnell, "Daily build and smoke test," IEEE software, 1996, vol. 13(4), p. 144.
- [10] E. van Veenendaal, "Standard glossary of terms used in software testing," International Software Testing Qualifications Board, 2010, pp. 1-51.
- [11] J. Humble and D. Farley, Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation. Addison-Wesley Professional, 2010, 512 p.
- [12] G. Mustafa, A. Shah, K. Asif, and A. Ali, "A Strategy for Testing of Web Based Software," Information Technology Journal, 2007, vol. 6(1), pp. 74-81
- [13] "Microsoft Corp. Guidelines for Smoke Testing," MSDN Library for Visual Studio 2008, accessed: November 13, 2018. [Online]. Available: [https://msdn.microsoft.com/en-us/library/ms182613\(v=vs.90\).aspx](https://msdn.microsoft.com/en-us/library/ms182613(v=vs.90).aspx).
- [14] M. Mozgovoy and E. Pyshkin, "Unity application testing automation with Appium and image recognition," In Itsykson V., Scedrov A., Zakharov V. (eds) Tools and Methods of Program Analysis. TMPA 2017. Communications in Computer and Information Science, vol 779. Springer, Cham, pp. 139-150.
- [15] Appium, project homepage, accessed: November 11, 2018. [Online]. Available: <http://appium.io>.
- [16] Calabash, project homepage, accessed: November 11, 2018. [Online]. Available: <http://calabash.sh>.
- [17] "World of tennis: Roaring '20s," project homepage, accessed: November 11, 2018. [Online]. Available: <http://worldoftennis.com/>.
- [18] "Automating user interface tests," accessed: November 13, 2018. [Online]. Available: <https://developer.android.com/training/testing/ui-testing/index.html>.
- [19] V. V. Helppi, "Using opencv and akaze for mobile app and game testing," (January 2016), accessed: November 13, 2018. [Online]. Available: <http://bitbar.com/using-opencv-and-akaze-for-mobile-app-and-game-testing>.
- [20] S. Kazmierczak, "Appium with image recognition," (February 2016), accessed: November 13, 2018. [Online]. Available: <https://medium.com/@SimonKaz/appium-with-image-recognition-17a92abaa23d/#.oez2f6hnh>.
- [21] "OpenCV Library," accessed: November 11, 2018. [Online]. Available: <http://opencv.org>.
- [22] "OpenCV: Template Matching," accessed: November 11, 2018. [Online]. Available: http://docs.opencv.org/master/de/da9/tutorial_template_matching.html.
- [23] G. Bradski and A. Kaehler, Learning OpenCV: Computer vision with the OpenCV library. O'Reilly Media, Inc., 2008.
- [24] R. Laganière, "OpenCV Computer Vision Application Programming Cookbook," 2nd ed., Packt Publishing, 2014.
- [25] M. Yamamoto, E. Pyshkin, and M. Mozgovoy, "Reducing False Positives in Automated OpenCV-based Non-Native GUI Software Testing," In Proceedings of the 3rd International Conference on Applications in Information Technology (ICAIT'2018), N. Bogach, E. Pyshkin, and V. Klyuev (Eds.). ACM, New York, NY, USA, 2018, pp. 41-45.
- [26] J. Canny, "A computational approach to edge detection," IEEE Transactions on pattern analysis and machine intelligence, no. 6, 1986, pp. 679-698.
- [27] GIMP, project homepage, accessed: November 11, 2017. [Online]. Available: <https://www.gimp.org/>.
- [28] "Imaging Web Demonstrations," Biomedical Imaging Group, accessed: November 13, 2018. [Online]. Available: <http://bigwww.epfl.ch/demo/ip/demos/edgeDetector/>.
- [29] N. Verma. Mobile Test Automation with Appium. Packt Publishing, 2017, 231 p.
- [30] C. Tao and J. Gao, "Cloud-Based Mobile Testing as a Service," International Journal of Software Engineering and Knowledge Engineering, 2016, vol. 26(1), pp. 147-152.
- [31] M. Linares-Vásquez, K. Moran and D. Poshyvanyk, "Continuous, Evolutionary and Large-scale: A New Perspective for Automated Mobile App Testing," 33rd IEEE International Conference on Software Maintenance and Evolution (ICSME'17), 2017.
- [32] Smartphone Test Farm, project homepage, accessed: November 11, 2018. [Online] Available: <https://openstf.io/>.
- [33] M. Mozgovoy and E. Pyshkin, "Mobile Farm for Software Testing," In Proceedings of the 20th International Conference on Human-Computer Interaction with Mobile Devices and Services Adjunct (MobileHCI '18), ACM, New York, NY, USA, 2018, pp. 31-38.
- [34] E. Pyshkin, "Designing human-centric applications: Transdisciplinary connections with examples," In Proc. of 2017 3rd IEEE International Conference on Cybernetics (CYBCONF), Exeter, UK, Jun 21-23, 2017, pp. 455-460.

Deriving Learning Strategies from Words Lists: Digital Dictionaries, Lexicons, Directed Graphs and the Symbol Grounding Problem

Jean-Marie Poulin and Alexandre Blondin Massé

Département d'informatique
Université du Québec à Montréal
Montréal, QC, Canada H3C 3P8

Email: {poulin.jean_marie, blondin_masse.alexandre}@uqam.ca

Abstract—We examine the structure of dictionaries, more specifically the interweaving of links that connect words through their definitions. With few exceptions, all the words used to construct dictionary definitions are defined somewhere else in the dictionary. All these references between words create a network of relations, thus making it possible to use graph theory for the study of dictionary structures. We propose using words learning as an investigative tool. For a given dictionary or lexicon, what would be the best strategy to learn all its headwords? To answer this question, we introduce a formal model and simple graph algorithms. We evaluate several different learning strategies by comparing their learning rate and their efficiency for 8 monolingual English-language dictionaries. It turns out that the most significant factor affecting the performance of learning strategies is their ability to break definitions circularity. In other words, the most effective learning strategies are the ones that break definition loops as quickly as possible. We show that a very simple algorithmic strategy, based solely on the vertices out-degree - the number of definitions in which lexemes participate - significantly improves the learning process when compared to psycholinguistic-based strategies. We also put forward that such an approach represents an efficient alternative for the construction of “word lists” used to teach foreign languages.

Keywords—Dictionaries; Lexicons; Learning Strategies; Word Lists; Graph theory.

I. INTRODUCTION

In this paper, which is an extended version of our earlier research presented at COGNITIVE 2018 [1], we examine the internal structure of dictionaries from a new and different perspective.

Whether in the form of clay tablets, papyrus, manuscripts, printed books, web pages or electronic tablets, dictionaries have existed for a very long time [2]. Since Antiquity, they have been commonly used as reference works in all areas of knowledge related to language. Even today, they are indispensable resources for reading, writing and translating texts, as well as for acquiring general knowledge.

With the advent of printing at the beginning of Renaissance, and even more so with the development of computers and the digital representation of knowledge in the 20th century, dictionaries underwent profound metamorphoses. In spite of this, dictionaries, lexicons and encyclopedias of all kinds still retain their relevance today. Open platforms, such as Wiktionary and Wikipedia, or Web versions of commercial dictionaries, such as Merriam-Webster [3] or Collins [4], are becoming ever more popular.

One of the key drivers of this success is undoubtedly the integration of hypertext and hyperlinks. These new technologies allow lexicographers to easily establish different types of links between words and concepts within a single publication, or even direct the user to external web resources. It then becomes possible to easily navigate from one word to another, without having to browse laboriously through the thousands of pages of a paper book. The way one uses dictionaries is thus profoundly modified. The relationships between the words become as important as the information about the words themselves.

Helped by developments in cognitive psychology and natural language processing, researchers have begun to question how these links between the words in dictionaries are organized. Are there invariants or schemes common to all dictionaries? This question was the main topic of several articles dealing with the structure of dictionaries.

In one of the first contributions on the subject, Clark [5] considered two special dictionaries: Longman's Dictionary of Contemporary English (LDOCE) [6] and Cambridge's International Dictionary of English (CIDE) [7]. LDOCE and CIDE have the particular feature of having a small *control vocabulary*, i.e., the number of words used in at least one definition is minimized. He showed that the words in the control vocabulary have distinctive properties: they are in general more abstract and their definition is longer and more complex than that of other words. Subsequently Steyvers and Tenenbaum [8] continued the analysis of the graphs associated with the WordNet Semantic Network [9] and Roget's Thesaurus [10].

Pursuing the same topic, a group of researchers have recently published a series of articles exploring the internal structure of dictionaries [11]–[15]. These works all use a formal model for the lexicon based on graph theory. This approach makes it possible to apply conventional graph processing algorithms to dictionaries and derive a wealth of linguistically relevant information. Their analysis of several English-language digital dictionaries shows that all of them have a common structure and contain the same basic components [14], i.e.,

- A Kernel**, which is a subset of words in the dictionary that can be used to define all other words. The kernel can in turn be subdivided into a series of subcomponents of varying size, consisting of clusters of closely related words.
- A Core**, which is the subcomponent of the kernel with the largest number of words. In all the digital dictionaries

studied, the core is considerably larger than the other subcomponents of the kernel.

A Minimum Grounding Set (MinSet), which consists of a subset of words smaller than the core, obtained by judiciously combining elements of the core and other subcomponents of the kernel. This is the smallest group of words that can be used to define all the other words in the dictionary.

In addition, it turns out that the kernel has some distinct psycholinguistic characteristics [15]:

- The words in the kernel are learned earlier, are more concrete and are used more frequently than other words in the dictionary.
- There is a strong correlation between the evaluated psycholinguistic variables age of acquisition, degree of abstraction and words frequency.
- Within the kernel itself, there is a marked gradation of these same measures when evaluating words from the kernel, the core and the minimum grounding set.

Among these observations, the key point is most likely the minimum grounding set (Minset) question. In [15] the authors establish a direct link between the Minset and the "*Symbol Grounding Problem*", first described by Harnad in [16]. This problem can be briefly summarized as follows: When one looks for the definition of a word in a dictionary, he sees that this definition is built using other words. If these other words are not known, one can, of course, look for them in the dictionary. But, at the risk of getting caught in an endless loop, the meaning of some words must be known and rooted in the sensorimotor experience: "[...] it can not be dictionary lookups all the way down!" [15].

The symbol grounding problem is especially acute when learning a new language. In addition to getting familiar with grammar and syntax, one must acquire vocabulary and learn enough words to be able to understand and be understood. It must be possible to associate the external form of a written or spoken word with its meaning in a given context. According to Schmitt [17]: "[The] form-meaning link is the first and most essential lexical aspect which must be acquired".

What is the best way to learn these new words? Are there special learning methods or preferred strategies? In many research works, such as Prince [18], Schmitt [19], and Joyce [20], the authors compare traditional approaches used by instructors to teach second language learners. In the first method called "L1 translation", new English words are explained to the student in its mother tongue (the first language or L1). For example, if the student is Spanish-speaking, the teacher would give him an explanation of the English word *cat* in Spanish, i.e., *gato* or *felino*. With the second approach, named either "L2 context" or "L2 definition", the student must deduce by himself the meaning of a new word using the context in which it was seen or through some other explanation in English (the 2nd language or L2). One could for example explain to Jacques, a French-speaking student, the English word *own* with a definition such as "to have or hold as property". Joyce [20] compares these two methods. The "L1 translation" method is preferred for students with lower levels of proficiency: "[...] L1 translations for intentional vocabulary learning is seen to be most effective for students at lower proficiency levels". On the other hand, the "L2 definition"

approach is the most effective for vocabulary development: "for the purposes of general language development, learning through an L2 definition is favored".

A simple language dictionary can thus be a surprisingly effective way to understand and memorize new words. But for this to be successful, there is however an important prerequisite. The learner must first master a *basic subset* of the words in the new language. Only in this way will he be able to profitably use a dictionary.

Let us illustrate this point with the student Jacques in the previous example. Suppose Jacques sees in an English text the word *own*, which he does not know. He therefore consults the Merriam-Webster and finds the definition: "to have or hold as property" [21]. Assuming that he already knows the meaning of the words *to*, *have*, *or*, *hold* and *as*, but not that of the word *property*, he looks further in the dictionary and finds a definition for the word *property*: "something that is owned by a person". Although he is familiar with the words *something*, *that*, *is/be*, *by* and *person*, this definition is not useful for him. He faces what we call a definition loop: he needs to know the meaning of *owned/own* to understand the meaning of *property*, whereas at the beginning he was trying to understand this same word *own*. This is the difficulty we previously mentioned, the "*Symbol Grounding Problem*". Dictionary definitions are not enough by themselves to learn new words. In order to break out of the Kafkaesque situation created by the definition loop, one of these 2 words has to be learned some other way. In this case, Jacques could ask his teacher to explain him either the word *own* or the word *property*.

These same issues lie at the heart of our questioning. We aim to study the close relationship between the structure of monolingual dictionaries and the way in which the words of a language can be learned. In the references cited above (eg: [15]), the authors analyze the internal structure of dictionaries using groups of words with specific properties in terms of graph structure or psycholinguistic characteristics. They evaluate the definitional relations between the words to determine if it is possible to discover clusters of words having properties related to symbol grounding. Our approach is complementary. We first develop word lists, called *learning strategies*, based either on sequences of words coming from existing psycholinguistic norms, or built using graph theory algorithms. We then study the behavior of our *learning strategies* with respect to a reference task: "learning" all the words in a dictionary. We determine how effectively the strategies manage to break the definition loops in the dictionaries, thus avoiding the *symbol grounding problem*.

The rest of this document is organised as follows. In Section II, after having introduced some linguistic terminology and recalled basic notions of graph theory, we propose a convenient way to represent a lexicon as a directed graph. In Section III, we describe the notion of *learning strategy*. We first look in more detail at the problem of symbol grounding. Next, we discuss the question of word lists, these teaching tools frequently used by language instructors. Subsequently, we propose a formal learning model as well as related algorithms used to evaluate the strategies efficiency, regarding their ability to perform the task of "learning" all the words of a lexicon. We outline in Section IV our experimental environment and document the source of the digital dictionaries and psycholinguistic norms. Then we describe in detail the two types of

learning strategies developed:

- the algorithmic strategies, built using graph theory algorithms;
- the psycholinguistic strategies, based on psycholinguistic norms, i.e., lists of words ordered according to specific psycholinguistic properties.

Section V is devoted to the actual description of the experiments carried out. We present how we collected data and measured the performance of the strategies used to learn whole dictionaries. Then we outline the results obtained in the form of tables and graphs and offer a quick analysis of the most significant observations. Section VI completes our presentation by highlighting important findings and suggesting other avenues for future research.

It should also be noted that this article is a free French to English translation, with several modifications, of the first author's Master thesis [22].

II. DICTIONARIES, LEXICONS AND GRAPHS

In order to clearly position the subject of our study, let us look at some common definitions of the word "dictionary".

"Dictionary: a reference source in print or electronic form containing words usually alphabetically arranged along with information about their forms, pronunciations, functions, etymologies, meanings, and syntactic and idiomatic uses"

Merriam-Webster [3]

"Dictionary: a book that gives a list of words in alphabetical order and explains their meanings in the same language, or another language"

Longman [23]

These descriptions are consistent with the traditional view that most people hold. However, if we study them in more details, a central element of the definition stands out: the term *words*. In the following example, we look at 2 sentences where *word* is used with two different meanings.

Example 1.

- "Parce que" is a French word that translates to "because". In this sentence *word* refers to the whole "Parce que" group.
- "Parce que" is written in two words. In this case, *word* corresponds directly to the usual definition of a *word* as suggested by Jackson [24]: "a sequence of letters bounded by spaces".

Here is another example, showing another aspect of the ambivalence of *word*.

Example 2.

- We found a cat on the porch.
- There are many cats in the neighborhood.

Here the problem is a variant of the one in the previous example. Are *cat* and *cats* two different *words*? If we apply once again the definition from Jackson [24], we can infer that they are different *words*. But the fact is that in both cases we clearly refer to the same "small domestic animal known for

catching mice" [3]. In sentence 2 b), the plural form *cats* is used to show that we are talking about several animals.

This kind of ambiguity thus represents an important problem for our intended automated dictionary processing: the term *word* is not precise enough. We need to find a better way to distinguish its various uses. This is the reason why we first introduce a more precise linguistic terminology, allowing us to mitigate the imprecision of the vocabulary. We then put this terminology to work in order to propose a more formal definition of a lexicon. Thereafter, after having recalled some elementary notions of graph theory, we describe a way to represent a lexicon as a directed graph.

A. Terminology

There is no consensus amongst the different schools of linguistics as to which terminology is to be preferred. In this section we therefore propose, in order to simplify the understanding of our document, a list of the basic linguistic terms needed to describe our formal model.

Lexicon:

From a linguistic point of view, what is the difference between a lexicon and a dictionary? In English, the term lexicon is a common synonym for dictionary. According to the Merriam-Webster [25] or the Handbook of Linguistics [26], it is a book containing a list of words, accompanied by their definition, presented in alphabetical order. In our article, we use the term lexicon in its strict linguistic sense, namely: "the theoretical entity that corresponds to all the lexical items of a language or of an individual, i.e., the mental lexicon" [27, p. 109]. Note that this definition refers to a "set of lexical items", and not to a "set of words". To further highlight the difference between a dictionary and a lexicon, let us add a few precisions:

- 1) A dictionary is a model, a particular representation of a language's lexicon. It emphasizes the descriptive aspect, the definition of the words.
- 2) A dictionary is usually presented in alphabetical order, while there is no such imperative for a lexicon.
- 3) In a lexicon, the relationships between words are as important as the words themselves: it is not just a sequential list of words. One can also see a lexicon as a web of words linked together by a complex network of various relationships.

Amongst the many different relationships that words can have between them, let us look at a few examples:

Example 3.

- In the sentence: "The cat is a domestic animal", CAT and ANIMAL are connected to each other by relations of hyponymy and hyperonymy. CAT is a hyponym of ANIMAL, while in the opposite direction, ANIMAL is a hyperonym of CAT.
- In the sentence "I saw a stray cat", the words CAT and STRAY are connected by another type of relationship. STRAY is a quality that is commonly applied to a CAT. However, the qualifier PURPLE, as in "I saw a purple cat", is mostly inappropriate for a cat, unless used in a very specific context, like in a comic book.
- If we define a CAT as a "small domestic animal known for catching mice" [3], the *words* SMALL, DOMESTIC,

ANIMAL, etc., have here a different relationship with CAT. They help to describe, to define what a CAT is.

Later on, we use this last type of relationship, termed a “definitional relation”, to explore the structure of lexicons.

Words, lexemes and others:

Let us look now at the different elements that make up our terminology. Figure 1 illustrates, in the form of an entity-relationship diagram, the reciprocal links that unite the linguistic terms required for our analysis. These terms, as well as the associated writing conventions, are strongly inspired by Polguère [27], [28].

word form: The Oxford Dictionary defines a **word form** as: “a (particular) form of a word; especially each of the possible forms taken by a given lexeme, distinguished by their grammatical inflections” [29].

Without going further into linguistic theory, we simply say that *cat* and *cats* are two different word forms of the lexeme CAT, both of which refer to the same lexical meaning <cat>. The terms “lexeme” and “lexical meaning” are defined later on.

Writing convention:

A **word form** is noted in italics, for example *cats*.

lexical item: A **lexical item** - or headword - is the basic unit of a lexicon, equivalent to an entry in a dictionary. “A lexical item, also called a lexical unit, is either a lexeme or a phrase. Each lexicon (lexeme or phrase) is associated with a given meaning [...]” [27, p. 69]

For example, “seat belt” and “cat” are both lexical items. “cat” is a simple lexical item consisting of a single word-form, equivalent to the lexeme CAT. On the other hand, “seat belt” is a compound lexical item comprising 2 associated word forms.

In our analysis however, we do not tackle the task of deciding whether a group of words corresponds to a compound lexical item or not. We believe that is a different, quite difficult problem, worthy of consideration on its own. We thus consider further on all word-forms as candidate lexemes.

lexeme: Let us examine again the two sentences in example 2. We understand that the two word-forms *cat* and *cats* both make reference to the same concept or idea: the lexical meaning <cat>. These word-forms are simply “inflected forms” of the same **lexeme** CAT. Here, “inflected form” refers to a morphological change, the addition of an affix or special ending to the final of a word (noun, pronoun, participle, adjective) according to its function in the sentence or proposition [30].

Polguère [27] defines a lexeme as a generalization of word-form linguistic signs: each lexeme of the language is structured around a meaning that can be expressed by a set of distinct word-forms. In other words, we can think of a lexeme as a way of identifying a precise lexical meaning, to which a series of grammatical variations represented by the different word-forms are associated. In the same manner, the word-forms *write*, *writes*, *written*, ... are different grammatical forms of the same lexeme WRITE (Spencer [31]).

Writing convention:

Lexemes are written in small capital letters, as in CAT. They can also be tagged, as in CHAT_N¹, where the exponent “1” indicates the result of disambiguation and the index “N” represents the part of speech.

lexical meaning: In this paper, we use the term **lexical meaning** to refer to the idea, the mental representation, to which a lexeme refers. “The lexical meaning refers to a mental concept that is associated with a lexical unit to express an idea” [32] The term lexical meaning can, according to the disciplines and the authors, be put in parallel with the related notions of concept: “Between all the individuals thus connected by the language, it will establish a kind of average: all will reproduce [...] the same signs united to the same concepts” [33], as well as category in philosophy and cognitive psychology, and signified in semiotics.

Writing convention:

The lexical meaning of a lexeme is noted with chevrons. For example, <cat> is the lexical meaning associated with the lexeme CAT.

lemma: According to Polguère, a **lemma** is the canonical word form used to designate a term [27, p. 135]. In French for example, we use the infinitive present to represent a verb, the masculine singular to represent a noun, etc. According to our nomenclature, we say that it is the word-form that has been chosen to identify one or more lexemes.

Writing convention:

1. A lemma is written in non-proportional font, for example CAT.
2. To distinguish the lexemes associated with the same lemma, we use an exponent between 1 and *n*, for example CHAT¹, CHAT², ..., CHAT^{*n*}.

In automatic language processing, lemmatization is the operation consisting of identifying the lemma that corresponds to the different word-forms of a lexeme. For example: the lemma GO is the result of the lemmatization of the word-forms *goes*, *went*, ... The distinction between the terms “lexeme” and “lemma” can sometimes seem difficult to establish. To help make them stand out, one only needs to remember that lexeme is rather related to meaning, to semantics, whereas lemma is related to form, to morphology.

part of speech: The parts of speech (POS) are classes that group together lexical items according to their grammatical properties [27]. For the purposes of our presentation, we consider that all lexemes are part of exactly one of the following 5 parts of speech: *noun*, *verb*, *adjective*, *adverb* and *stop word*. The first four classes - *noun*, *verb*, *adjective* and *adverb* - group the vast majority of lexemes. The fifth class, *stop word*, groups all the other lexemes whose semantic value is poorer.

Writing convention:

The part of speech of a lexeme or a lemma is represented by a coding label: “_N” for name, “_V” for verb, “_A” for adjective, “_R” for adverb and “_S” for stop word. For example, CAT_N indicates that the lexeme CAT is a noun.

In natural language processing (NLP), the term “lemma-

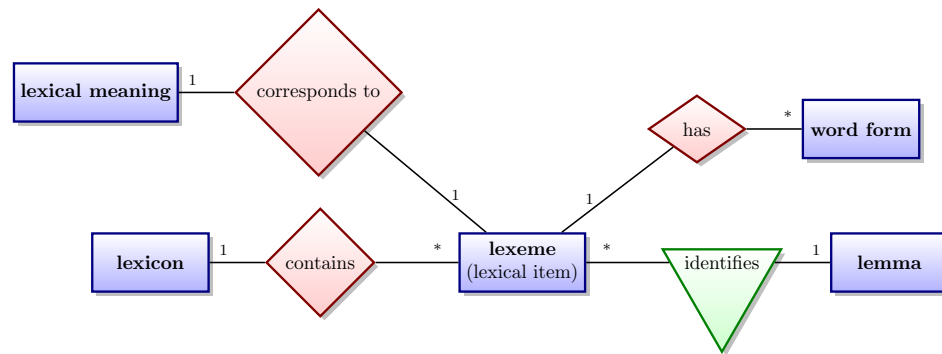


Figure 1. Entity-relationship diagram of linguistic terms (simplified)

tization” is used to refer to this process of identifying the part of speech to which the lexemes of a sentence or text belong.

Polysemy and Disambiguation:

We previously mentioned that a given lemma can correspond to more than one lexeme. In this regard, linguistic researchers usually make the distinction between two different situations [34] :

- *homonymy*, when the lexemes are of different etymological origin.
- *polysemy*, when the lexemes refer to different meanings of the same word

The next example, extracted from Jurafsky and Martin’s work [35], illustrates in a classical way how a “word” can have many different meanings:

Example 4.

- “Instead, a *bank* can hold the investments in a custodial account in the client’s name.”
- “But as agriculture burgeons on the east *bank*, the river will shrink even more.”
- “The *bank* is on the corner of Nassau and Witherspoon.”

To understand these sentences, one has to choose amongst some possible meanings for the lexeme BANK which one is the most appropriate, for example:

BANK_N¹: “financial institution”,
BANK_N²: “building belonging to a financial institution”,
BANK_N³: “sloping mound”,

In sentences (a) and (c), the context is quite different. It is relatively easy to disambiguate BANK. Sentence (a) is about investment, account and client, so BANK_N¹ is the most appropriate. In (b) however, BANK_N³ is the most relevant since we are in the context of agriculture, river, etc. Given that the semantic domain is downright different, we can easily identify them as homonyms. On the other hand, sentence (c) is more difficult to analyze. We do not have many clues from the context to guide our choice. One must know or figure out that Nassau and Witherspoon are street names and then infer

that we are talking about a building, therefore the branch of a bank. BANK_N² and BANK_N¹ are thus polysemous.

This complex process of discriminating the meaning of words is called “Word Sense Disambiguation” (WSD) or simply disambiguation. For a human, the distinction is made naturally, without apparent effort. It is however much more difficult for an algorithm or computer program: “The reason that lexical polysemy causes so little actual ambiguity is that, in actual use, context provides information that can be used to select the intended sense. Although contextual disambiguation is simple enough when people do it, it is not easy for a computer to do” [36] According to Corrêa, Lopes and Amancio, the question of lexical disambiguation in artificial intelligence even remains an unresolved problem in 2018 [37]. For several authors, it is considered an AI-complete problem. In other words, by analogy with NP-complete problems in complexity theory, it is a problem as difficult as the creation of a real artificial intelligence [38], [39].

There is thus no cost-efficient and reliable way to disambiguate the meaning of words in a sentence. However, when they build or revise dictionaries, lexicologists usually order word senses according to their usage frequency, starting from the most frequently used : CIDE: [7, p. ix], LDOCE: [40], WORDSMYTH: [41]. Thus, by simply using the definitions order, “the heuristic of the first sense” generally gives satisfactory results. This method is still a baseline difficult to surpass: “The first sense heuristic [...] outperforms many of these systems which take surrounding context into account” [42]. For these reasons, as well as for the sake of simplicity, we use the first sense heuristic as a disambiguation method in this work.

B. Formal definition of a lexicon

As we have seen above, a lexicon can be described from a linguistic point of view as a set of lexemes accompanied by their definitions and any other information necessary for their use [26].

However, for our analysis, we need to go further in terms of mathematical formalism. Proceeding by successive refinements, we propose in this section the formal definition of a *complete lexicon*.

Definition 1 (Lexicon). A *lexicon* is a quadruple $X = (A, \mathcal{P}, \mathcal{L}, \mathcal{D})$, where:

- (i) \mathcal{A} is an alphabet, whose elements are called letters.
- (ii) $\mathcal{P} = \{N, V, A, R, S\}$ is a non-empty set of elements called *part of speech* (POS). The elements correspond to the 5 parts of the speech described earlier.
- (iii) \mathcal{L} is a finite set of triplets $\ell = (w, i, p)$, called *lexemes* and denoted $\ell = w_p^i$, where $w \in A^*$ is a word form, $i \geq 1$ is an integer, and $p \in \mathcal{P}$. We then say that (w, i, p) is the i -th sense of the tagged word form (w, p) :
 - If there is no $(w, i, p) \in \mathcal{L}$ with $i > 1$, then $w_p^1 \equiv w_p$ and (w, p) is *monosemic*. Moreover, if all $(w, i, p) \in \mathcal{L}$ are monosemic, then we say that X is monosemic.
 - If there exists a $(w, i, p) \in \mathcal{L}$ with $i > 1$, we say that (w, p) and X are *polysemic*.
 - To make the numbering consistent, we assume that if $(w, i, p) \in \mathcal{L}$ and $i > 1$, then $(w, i-1, p) \in \mathcal{L}$ is also true.
 - If $p = S$, then $\ell = w_s^i$ is called a *stop lexeme*.
- (iv) \mathcal{D} is a function that associates with each lexeme $\ell \in \mathcal{L}$ a finite sequence $D(\ell) = (d_1, d_2, \dots, d_k)$, where $d_i \in A^*$ for $i = 1, 2, \dots, k$. It is called the definition of ℓ .

We can see in Example 5 a polysemic lexicon.

Example 5. Let $X = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ be a lexicon such that:

- $\mathcal{A} = \{a, b, \dots, z\}$
- $\mathcal{P} = \{N\}$, where N shows that the part of the speech is a NOUN,
- \mathcal{L} and \mathcal{D} are as defined in Table I.

TABLE I. Lexemes and definitions of a polysemic lexicon

ℓ	$D(\ell)$
FRUIT _N ¹	(plant, part, that, has, seed, and, edible, flesh)
FRUIT _N ²	(the, result, of, work, or, action)
FLESH _N ¹	(the, edible, part, of, a, fruit, or, vegetable)
FLESH _N ²	(the, part, of, an, animal, used, as, food)
SEED _N ¹	(the, small, part, of, a, plant, from, which, a, new, plant, can, develop)

Definition 2 (Lemmatized lexicon). Let $\text{lemma}(w)$ be a function that associates to a word-form $w \in A^*$ its lemma. If we replace in Definition 1 (iv) $D(\ell) = (d_1, d_2, \dots, d_k)$ by $D(\ell) = (\text{lemma}(d_1), \text{lemma}(d_2), \dots, \text{lemma}(d_k))$, then $D(\ell)$ is called a *lemmatized definition* of ℓ .

We then say that X is a *lemmatized lexicon*.

Definition 3 (Tagged Lexicon). If we replace in Definition 2 (iv) the condition $d_i \in A^*$ with $d_i \in \mathcal{A}^* \times \mathcal{P}$, then $D(\ell)$ is called a *tagged definition* of ℓ .

We then say that X is a *tagged lexicon*. Example 6 shows such a lexicon.

Example 6. Let $X = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ be a lexicon such that:

- $\mathcal{A} = \{a, b, \dots, z\}$
- $\mathcal{P} = \{N, V, S\}$, where $N \rightarrow \text{NOUN}$, $V \rightarrow \text{VERB}$, $S \rightarrow \text{STOP}$
- \mathcal{L} and \mathcal{D} are as defined in Table II.

TABLE II. Lexemes and definitions for a tagged lexicon

ℓ	$D(\ell)$
HAVE _V	(to _S , own _V , or _S , possess _V)
OWN _V	(to _S , have _V , in _S , your _S , possession _N)
POSSESS _V	(to _S , have _V , in _S , its _S , possession _N , to _S , own _V)
POSSESSION _N	(having/have _V , or _S , owning/own _V , something _S)

Definition 4 (Disambiguated lexicon). If we replace in Definition 3 (iv) the condition by $d_i \in \mathcal{L}$, then $D(\ell)$ is called a *disambiguated definition* of ℓ .

We then say that X is a *disambiguated lexicon*.

Definition 5 (complete lexicon). Finally, if X is a *disambiguated lexicon* such that for every ℓ that is not a stop lexeme there is a $D(\ell)$, we then say that X is a *complete lexicon*.

Example 7. Let $X = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ be a lexicon such as:

- $\mathcal{A} = \{a, b, \dots, z\}$
- $\mathcal{P} = \{N, V, S\}$, where $N \rightarrow \text{NOUN}$, $V \rightarrow \text{VERB}$, $S \rightarrow \text{STOP}$
- \mathcal{L} and \mathcal{D} are defined in Table III.

TABLE III. Lexemes and definitions for a complete lexicon

ℓ	$D(\ell)$
HAVE _V	(TO _S , OWN _V , OR _S , POSSESS _V)
OWN _V	(TO _S , HAVE _V , IN _S , YOUR _S , POSSESSION _N)
POSSESS _V	(TO _S , HAVE _V , IN _S , ITS _S , POSSESSION _N , TO _S , OWN _V)
POSSESSION _N	(having/HAVE _V , OR _S , owning/OWN _V , SOMETHING _S)

C. Graphs

In this section, we give a brief overview of the mathematical model used for our analysis of the structure of lexicons: the graph theory. But first, let us introduce the notion of semantic network.

For many authors specializing in artificial intelligence, a semantic network is an especially useful form of knowledge representation [43]–[45]. Lehmann gives a very concise definition: “A semantic network is a graph of the structure of meaning” [46]. In its traditional form, a semantic network represents objects in the form of nodes, connected to each other by links, which are optionally labeled. Figure 2 provides an example of a simple semantic network. Nodes and arrows represent a subset of a database of free associations [47]. In this study, the authors asked participants, after showing them a word, to name the first word that spontaneously came to their mind. For example, in the diagram in Figure 2, “volcano” is connected to “explode” by an arrow. This means that several participants spontaneously associated the word “explode” with the word “volcano” when the latter was used as a primer.

Using the same type of representation, one can easily imagine representing a lexicon as a graph where the lexemes are displayed as nodes and the relations between the lexemes are indicated by links between the nodes. As an example, let us go back to the definition of the lexeme HAVE_V in example 7:

$$D(\text{HAVE}_V) = (\text{TO}_S, \text{OWN}_V, \text{OR}_S, \text{POSSESS}_V)$$

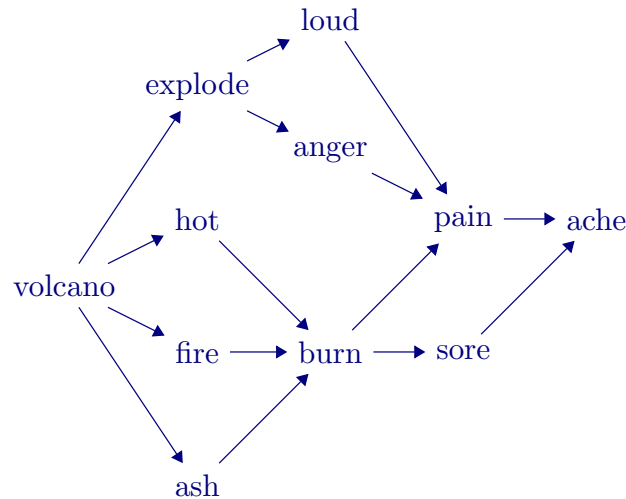


Figure 2. Association network [8].

Figure 3 represents this same definition of $HAVE_v$ as a semantic network.

This form of representation is quite similar to the way Bondy and Murphy introduce the notion of graph [48, p. 1], i.e., “[...] a diagram consisting of a set of points together with lines joining certain pairs of these points”. But to properly represent definitional relations between lexemes, we must be more specific and introduce the notion of directed graph.

Definition 6 (Directed graph). A *directed graph* D – digraph – is an ordered pair (V, A) where:

- (i) V is a finite set of *vertices*,
- (ii) $A \subseteq V \times V$ is a finite set of elements called *arcs*,

Note: If $v_1, v_2 \in V$, then $(v_1, v_2) \in A$ does not imply that $(v_2, v_1) \in A$.

Example 8. Let $D = (V, A)$ be a directed graph with

$$V = \{v_1, v_2, v_3, v_4\},$$

$$A = \{(v_2, v_1), (v_3, v_1), (v_1, v_2), (v_3, v_2), (v_4, v_3), (v_2, v_3), (v_2, v_4), (v_1, v_4), (v_3, v_4)\}$$

Figure 4 is a visual representation of the digraph D .

From this definition of a directed graph, we derive the following related notions:

degree

Let $D = (V, A)$ be a directed graph. For $u, v \in V$, u is a *predecessor* of v if $(u, v) \in A$. The set of predecessors of v is written $N^-(v)$. The number of predecessors of v is called the *in-degree* of v , represented by $\deg^-(v)$. In the same manner, we say that v is a *successor* of u if $(u, v) \in A$ and that the set of successors of u is denoted $N^+(u)$. In this case $\deg^+(u) = |N^+(u)|$ is called the *outer degree* of u .

circuit

A finite sequence $p = (v_1, v_2, \dots, v_k) \in V^k$ is called a

path of D if $(v_i, v_{i+1}) \in A$ for $i = 1, 2, \dots, k-1$. If in addition $v_1 = v_k$, then p is called a *circuit*.

feedback vertex set

A *feedback vertex* of D is a subset $U \subseteq V$ of vertices such that, for any set of vertices c forming a circuit in D , the set $U \cap c$ is non-empty [49]. That is, U covers all circuits of D . The *minimum feedback vertex set (MFVS) problem* consists in finding in a graph a feedback vertex set of size as small as possible. For a general graph, it is an NP-hard problem, namely that there is no algorithm to solve this problem in polynomial time unless $P = NP$ [50]. However, by using combinatorial operators and linear programming techniques [51], [52], Vincent-Lamarre et al. [15] have succeeded in solving the problem for the smallest lexicons they considered and in finding a good approximation for the other ones.

strongly connected component

For $u, v \in V$, we write $u \rightarrow v$ if there exists a path from u to v and we write $u \leftrightarrow v$ if both $u \rightarrow v$ and $v \rightarrow u$ hold. A *strongly connected component (SCC)* is a subgraph of D induced by an equivalence class of the relation \leftrightarrow over V . In other words, when it is possible to move from a vertex u to a vertex v in a strongly connected component, it is also possible to go in the opposite direction from the vertex v to the vertex u . Moreover, since \leftrightarrow is an equivalence relation and in particular, transitive, the induced subgraph will be of maximal size [11].

D. Lexicons and Associated Graphs

Directed graphs are especially suitable for representing the relations between the lexemes of a lexicon. For our analysis of the structure of lexicons, we consider only the definitional relations of the type: lexeme l “is part of the definition” of lexeme l' .

We represent a lexicon using the following conventions:

- The vertices of the graph correspond to the lexemes.

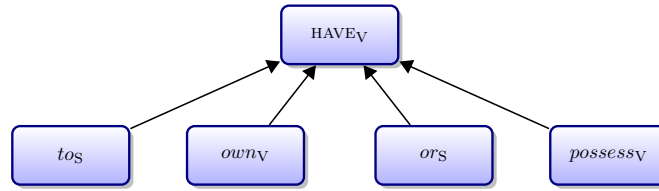
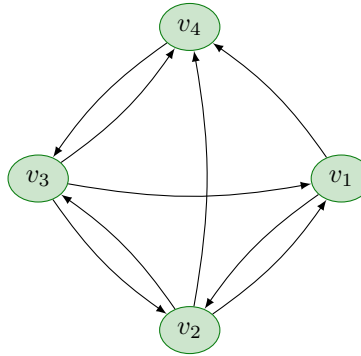


Figure 3. Semantic Network Representing Definitional Relationships

Figure 4. Digraph D

- The arcs between the vertices correspond to the relations between the lexemes. For example, if an arc goes from vertex l to vertex l' , it means that lexeme l is part of the definition of l' .
- With regard to stop lexemes, we consider that their lexical value is very low compared to lexemes of other parts of speech (noun, verb, adjective and adverb). We do not represent them in the associated graphs and we do not take them into account in our analysis. This way of doing things is used very often in NLP [35], in information research (RI) [53], and in data mining [54].

More formally, we define an “associated graph”, as follows.

Definition 7 (Associated graph).

Let $X = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ be a complete lexicon. Then $G(X)$ is X ’s associated graph if:

- $G(X) = (V, A)$ is a directed graph
- $V = \mathcal{L}$
- If $\ell \in D(\ell')$ and ℓ is not a stop lexeme, then $(\ell, \ell') \in A$

The following example 9 shows the graph associated with the small lexicon X_{small} , containing 4 vertices – 4 lexemes – and 9 arcs – 9 definitional relations –.

Example 9.

Let $X_{small} = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ be a complete lexicon where \mathcal{L} and \mathcal{D} are shown in Table IV.

Figure 5 illustrates the graph corresponding to the lexicon X_{small} .

Example 10.

Figure 6 shows the graph associated with the larger $X_{large} = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ lexicon, comprising 40 vertices and

TABLE IV. Complete lexicon

ℓ	$D(\ell)$
HAVE _V	(TO _S , OWN _V , OR _S , POSSESS _V)
OWN _V	(TO _S , HAVE _V , IN _S , YOUR _S , POSSESSION _N)
POSSESS _V	(TO _S , HAVE _V , IN _S , ITS _S , POSSESSION _N , TO _S , OWN _V)
POSSESSION _N	(<i>having</i> /HAVE _V , OR _S , <i>owning</i> /OWN _V , SOMETHING _S)

123 arcs.

III. LEARNING STRATEGIES

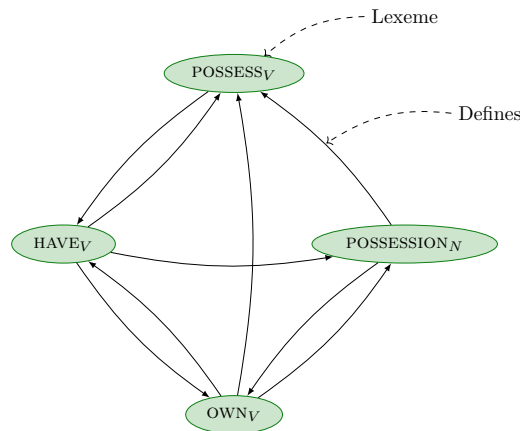
In this section, we examine the relationship between the learning of new words and the structure of dictionaries.

First, we look into what is implied by the phrase “learning new words”. We seek to understand how we learn to associate a linguistic token, whether read or heard, with a meaning. For this purpose, we reexamine the symbol grounding problem [16] and the pedagogical approach traditionally used to mitigate this difficulty: the construction of word lists.

In a second step, we propose a high-level model to represent the process of learning new words. After having formally defined what “learning a new word” means in our context, we propose different algorithms to simulate this behaviour.

A. Learning new words

In this section, we address the issue of vocabulary acquisition, particularly in the context of second-language learning. First, we seek to identify the main difficulties encountered when using a monolingual dictionary to learn the meaning of the new words encountered.

Figure 5. Graph associated with the lexicon X_{small} Symbol Grounding Problem:

In several articles dealing with this matter, Harnad analyzes the problem of grounding symbols, the famous Symbol Grounding Problem [16], [55], [56]. Without going into the details of linguistics and cognitive science, this question can be summarized as follows: Where does the meaning of words come from? How is it that a word we know usually conjures up something specific? According to Harnad, this is because the words are grounded in a sensorimotor way:

“How are word meanings grounded? Almost certainly in the sensorimotor capacity to pick out their referents.” [56]

However, it is clear that the learning of new words does not occur in the same way for a young child assimilating the first basics of his mother tongue as for an adult studying a new language.

When a second-language learner encounters a word he does not know, one way around this difficulty may be to consult a dictionary to find the definition of the unknown word. If everything goes well, the definition allows him to “learn” the new word and memorize it. Let us illustrate this situation with an example from [16]:

- (1) Suppose a learner already knows the word *horse*, which is well grounded in his sensorimotor experience. He can easily recognize a horse if he sees one.
- (2) Let’s also suppose that *striped* is known in the same manner
- (3) He would then presumably be able to identify a zebra if he sees one, using only a simple definition such as “*striped horse*”. He could associate the symbol – the word *zebra* – with the animal that looks like a horse and that is striped.

But things get more complicated if there are too much words in the definition that he does not know. In the article of Blondin Massé *et al*, the authors describe the uncomfortable situation where one would endlessly run through the dictionary, going from unknown words to other unknown words, without hope of arriving at understanding of the words and of their definitions [11]. Therefore, for the definition of a word in a dictionary to be understandable and useful, a sufficient number

of words must already be “grounded”, that is, they must mean something more than abstract forms on paper or on a screen. We do not study further how words are actually grounded in sensorimotor experience. We keep in mind that if enough words in the definition are known and well grounded, one can learn a new word and ground it in turn.

Minimum Grounding Set:

We now examine how our formal model for lexicons and associated graphs reflects the symbol grounding issue.

First, assume that we can learn a new lexeme only if we already know all the lexemes that appear in its definition. We can then define a grounding set as a subset of lexemes allowing us to learn all the other lexemes in this lexicon.

Definition 8 (Grounding set). Given

- (a) $X = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ a complete lexicon,
- (b) $G(X) = (V, A)$ its associated graph,
- (c) $U \subseteq V$ a subset of V ,
- (d) L a function defined by $L(U) = U \cup \{v \in V \mid N^-(v) \subseteq U\}$,

if there is a $k \in \mathbb{Z}^+$ such that $L^k(U) = V$, we then say that U is a *grounding set* of X and that \mathcal{L} is *k-reachable*.

Looking again at lexicon X_{large} in Figure 6, we can use this definition to validate if a subset of the vertices of X_{large} is a grounding set.

Example 11.

Let us use a starting subset $U = \{ \text{HAVE}_V^1, \text{PLACE}_N^1, \text{POSSESSION}_N^1, \text{QUALIFY}_V^1, \text{REFER}_V^1, \text{STATE}_N^1, \text{THING}_N^1 \}$.

If we recursively apply the previously defined function L , we get:

$$\begin{aligned}
 L^0(U) &= U \\
 L^1(U) &= L^0(U) \cup \{ \text{PARTICULAR}_A^1, \text{POSITION}_N^1, \text{OWN}_V^1 \} \\
 L^2(U) &= L^1(U) \cup \{ \text{POSSESS}_V^1, \text{SOMETHING}_N^1 \} \\
 L^3(U) &= L^2(U) \cup \{ \text{CONDITION}_N^1 \} \\
 L^4(U) &= L^3(U)
 \end{aligned}$$

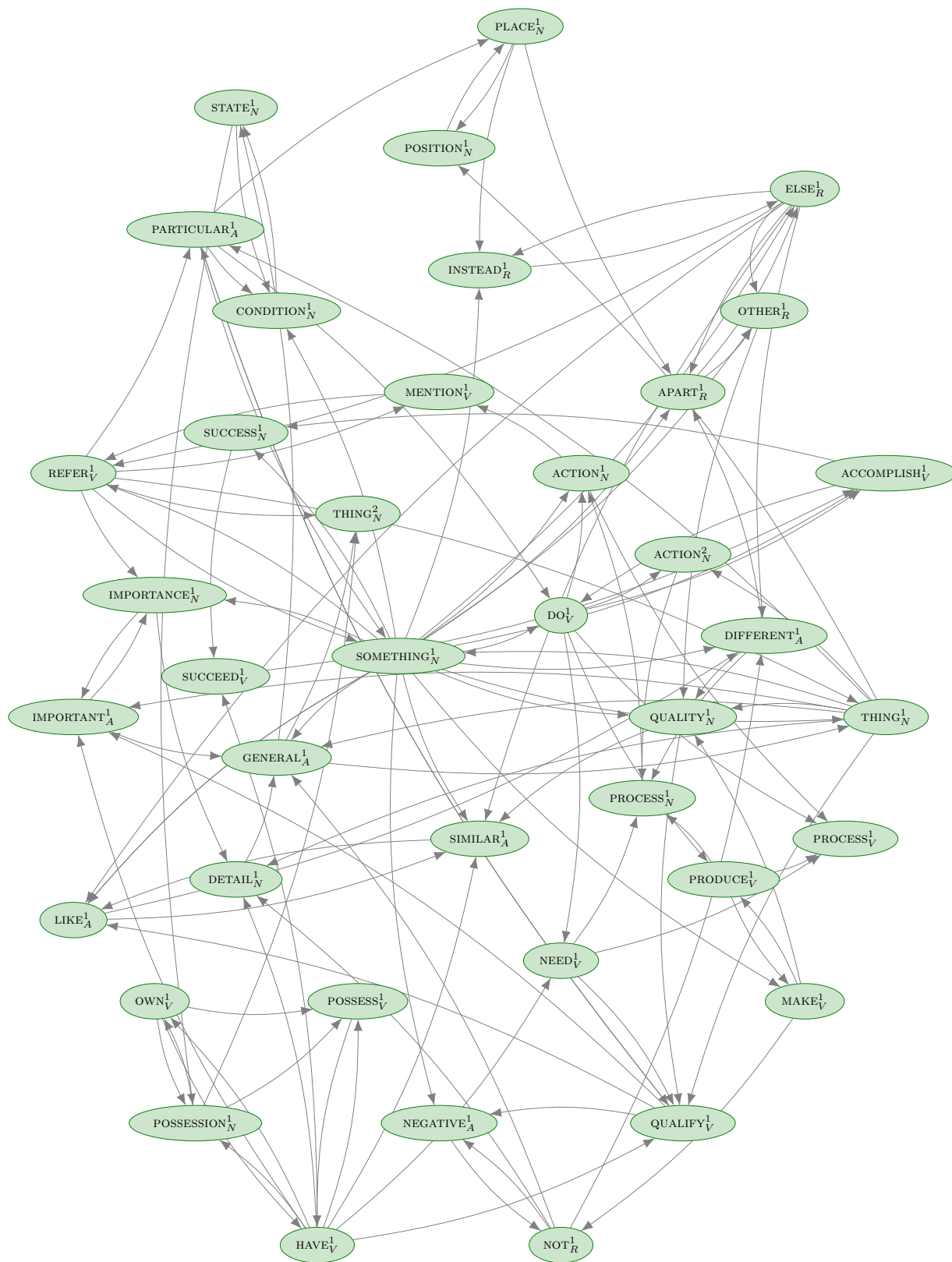






Figure 6. Graph associated with lexicon X_{large}

In Figure 7, the elements of the sets $L^0(U)$, $L^1(U)$, $L^2(U)$ and $L^3(U)$ are respectively marked with the symbols , ,  and .

For example, we can see that the lexeme OWN_V^1 is 1-reachable since it can be learned from the lexemes POSSESSION_N^1 and HAVE_V^1 . Similarly, the lexeme POSSESS_V^1 is 2-reachable since it can be learned from the lexemes POSSESSION_N^1 and HAVE_V^1 , and OWN_V^1 . Moreover, since we have $L^4(U) = L^3(U)$, there is no way we can learn additional lexemes and U is not a grounding set of the X_{large} lexicon.

Blondin Massé *et al.* have shown that there is also an exact correspondence between the grounding sets of a lexicon X and the feedback vertex sets of the associated graph $G(X)$ [11]. So, if U is a grounding set of $G(X)$, it means that we can learn by definition all the other lexemes of X , that is $V \setminus U$. As explained earlier in Section II-C, the calculation of a minimum feedback vertex set is, in general, an NP-hard problem. However, as the article by Vincent-Lamarre *et al.* [15] demonstrates, it is possible to use algorithms and linear programming techniques to calculate an exact solution or at worst to find a close-enough approximation. To illustrate the calculation of minimal grounding sets, let us look again at examples of complete lexicons presented in Section II-D.

Example 12.

For the trivial lexicon X_{small} from Figure 8, one can easily find by trial and error a minimum feedback vertex set, for instance: $\{\text{HAVE}_V^1, \text{POSSESSION}_N^1\}$.

Example 13.

On the other hand, for the X_{large} lexicon, which is still diminutive compared to a real-life dictionary, we find that the “manual” method is not adequate for finding a minimum grounding set. Figure 9 illustrates a minimum grounding set obtained using the method described in Vincent-Lamarre *et al.* [15].

$\{\text{ACCOMPLISH}_V^1, \text{HAVE}_V^1, \text{IMPORTANT}_A^1, \text{LIKE}_A^1,$
 $\text{MAKE}_V^1, \text{PLACE}_N^1, \text{POSSESSION}_N^1, \text{QUALIFY}_V^1,$
 $\text{REFER}_V^1, \text{STATE}_N^1, \text{THING}_N^1, \text{NOT}_R^1, \text{ELSE}_R^1\}$

B. Word lists

Let us look at the connection that can be established between the notions of symbol grounding and minimal grounding set, and the techniques used for teaching languages.

The importance given to vocabulary teaching in second language classes has varied over the years, following the evolution of theories and approaches in language didactics [57]. But the fact remains that for students, the acquisition of a large vocabulary is essential for attaining proficiency in a language. Teachers and researchers in applied linguistic have thus long sought ways to facilitate the learning of new words for their students. In this context, one can understand their interest in word lists.

Word lists are of word groupings representative of a specialized field or a language that students must master as early as possible to become autonomous in their study. They then have a base of known words allowing them to independently use dictionaries or other tools to help learning. According to Nation, “Word lists lie at the heart of good vocabulary course design” [58].

In the 1930s, Charles Ogden first introduced his “Basic English”, a version of English with simplified grammar and vocabulary [59]. Basic English was to become, according to Ogden, a universal language, somewhat like Esperanto. To facilitate the learning of this basic English, several lists of words - between 850 and 2000 words - were subsequently built [60].

In the 1950s, West proposed its General Service List (GSL), containing about 2000 words frequently used in English [61]. The GSL has since become a key reference: “There has been no comparable replacement for the GSL up to now” [62].

More recently, Brezina and Gablasova [63], as well as Browne [64] both proposed an improved version of the GSL, named in both cases the New General Service List (NGSL). Browne also suggests 3 additional lists to complement the NGSL [65]:

- The “New Academic Word List” (NAWL);
- The “TOEIC Service List” (TSL);
- The “Business Service Lists” (BSL).

But how are these lists constructed? The most commonly used method is to count the relative frequency of words in a collection of relevant documents and then classify those words

in a list according to their frequency and their importance for the author. In a recent publication, Nation presents a detailed description of list construction techniques using corpora [58].

In this article we propose a different approach, never used before as far as we know. With this new method, we use a lexicon and simple graph theory algorithms to efficiently build word lists. To accomplish this, we first represent the lexicon as a directed graph and then use graph algorithms to identify a list of words allowing us to effectively “learn” all the other words of the lexicon.

C. Learning model

In an article by Picard *et al.* [13], the authors put forward the hypothesis that there are two ways to learn new words or new lexical meanings: verbal instruction and direct sensorimotor induction.

We rely on this premise to build our formal model of learning. We say that a new lexeme can be learned in two different ways:

Direct learning: With this approach, lexeme and lexical meaning are directly connected through sensorimotor experience. For example, during a visit to a farm, someone could explain to a child that the animal in front of him is called a “horse”.

To keep our model simple, we do not concern ourselves about the way this link is established or what is going on at the mental and sensorimotor levels. We stick to the fact that it is a complex operation, which often requires the intervention of a person or some other entity to clarify the matter. We have to get out of the pure “world of words”, so we consider it to be a relatively costly process.

Definition learning: In this case, some lexical information is used to establish the link between the meaning and the lexeme; for example, a student searches a dictionary to find the definition of a zebra.

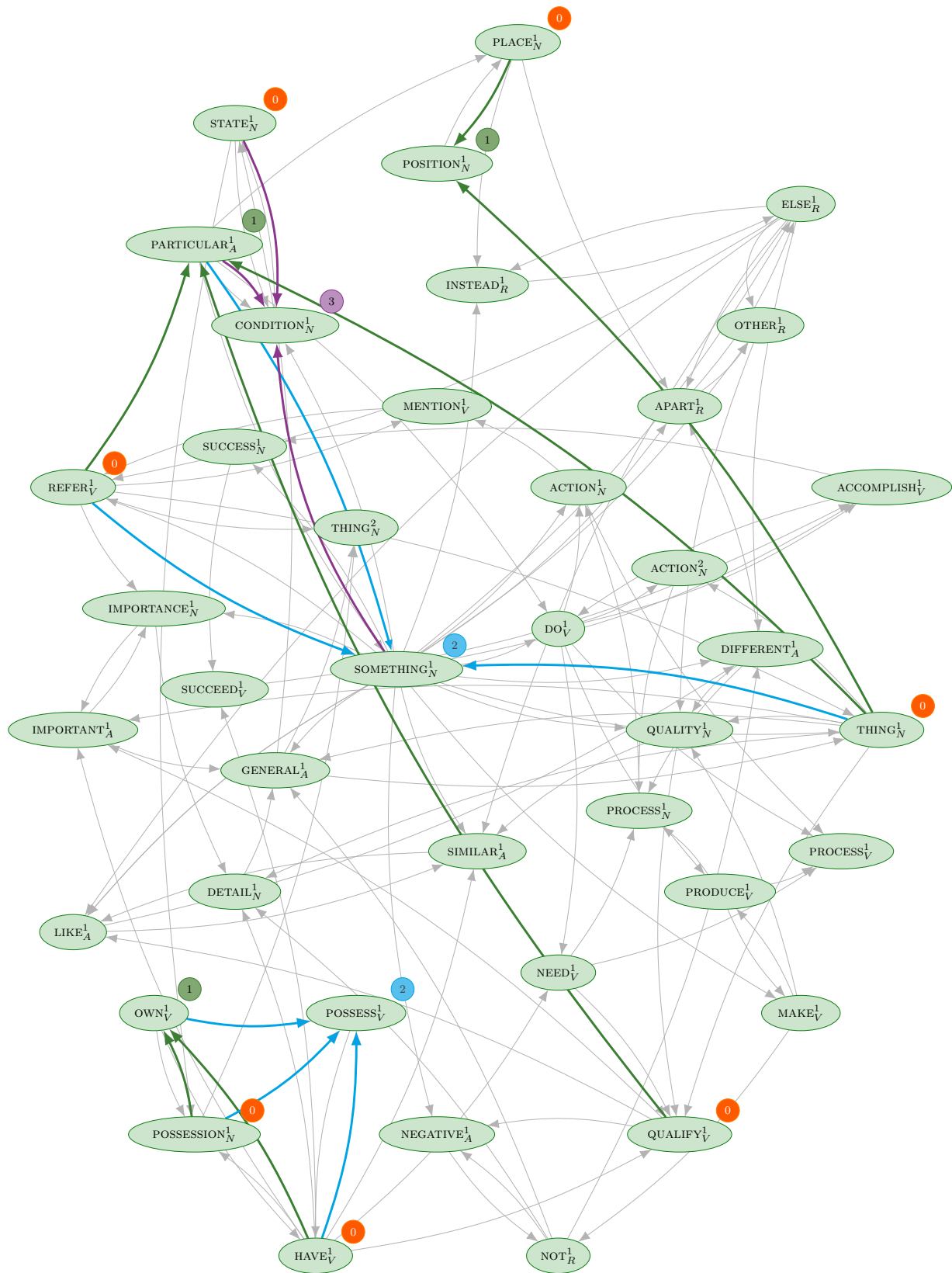


Figure 7. Graph associated to lexicon X_{large}
(Lexemes are marked according to their k -reachability from U).

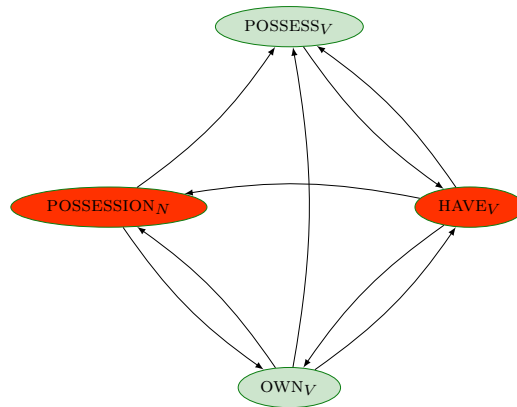


Figure 8. Graph associated to lexicon X_{small}
(Lexemes in the minimum feedback vertex set are marked in red).

We assume that this form of learning is much less expensive than direct learning. It does not require the intervention of people, nor the participation of a third party to provide explanations; we remain in the exclusive sphere of words and meanings.

Nevertheless, in order to avoid falling into the trap of symbol grounding, we need to assume that a lexeme can only be learned by definition if it is completely defined, that is all lexemes in its definition are already known.

In our model, we use a monolingual lexicon as our external data source.

Within this model, our learning objective is stated as follows. Starting from an initial situation where we do not know the meaning of any lexeme, we aim to learn the meaning of all the lexemes of a lexicon. To do this, the learning process involves learning the lexemes one by one according to the following rules:

- 1) If a lexeme in the lexicon is unknown, but all the elements in its definition are known already, we learn it *by definition*.
- 2) Otherwise, we learn *directly* the next lexeme indicated by the learning strategy.
- 3) Repeat the previous steps until the entire lexicon is learned.

Learning Strategy:

Let us now formally define a learning strategy.

Definition 9 (Learning Strategy). Let $X = (\mathcal{A}, \mathcal{P}, \mathcal{L}, \mathcal{D})$ be a complete lexicon.

- (i) A *learning strategy* S is an ordered sequence of elements from \mathcal{L} .
- (ii) If the sequence S viewed as a set is a grounding set of X , we say that S is *exhaustive*.
- (iii) Otherwise, we say that S is *non exhaustive*.

In other words, a learning strategy for a lexicon is simply a list of lexemes in that lexicon sorted in the order in which they are to be learned. As we will see in the next section, this list can be derived from an external word list, for example

the Brysbaert and New [66] usage frequency list, or it can be determined using an algorithm. It is an exhaustive strategy if it allows us to learn all the lexemes in the lexicon.

Taking into account the two learning ways described above, we intuitively find that the learning effort for a strategy will be minimized if it requires to learn directly as few lexemes as possible. Without loss of generality, we further assume the cost of learning directly a lexeme to be 1 and 0 for learning by definition. We also say that a given strategy S_1 is more efficient than strategy S_2 if S_1 allows to fully learn the lexicon at a lower cost than S_2 .

Learning Algorithms:

We now expose the 3 algorithms that will let us calculate the cost of a learning strategy and determine if it is exhaustive.

Algorithm 1: Partial learning cost

The PARTIALCOST function computes the actual cost attributed to a strategy. As we mentioned before, some strategies, deemed non-exhaustive, fail to completely learn a lexicon. If so, PARTIALCOST calculates the cost so far and returns the portion of the lexicon that could not be learned. Otherwise, if the strategy is exhaustive, the cost returned corresponds to the total cost and there are no more lexeme to learn. Incidentally, this also allows us to verify if a strategy is exhaustive or not.

The function PARTIALCOST() accepts the following parameters:

- S , a learning strategy.
- X , a lexicon.

It returns as result the couple $(cost, X')$, where:

- $cost$ is the cost incurred by the strategy S for learning lexicon X ,
- X' is the remaining portion of X that could not be learned with S . X' can be used to determine if S is exhaustive:
 - If lexicon X' is empty, then the strategy S is exhaustive and $cost$ is equivalent to the total cost.
 - If lexicon X' is not empty, then strategy S is non-exhaustive. We must then use a fallback strategy to completely learn the lexicon and get the total cost.

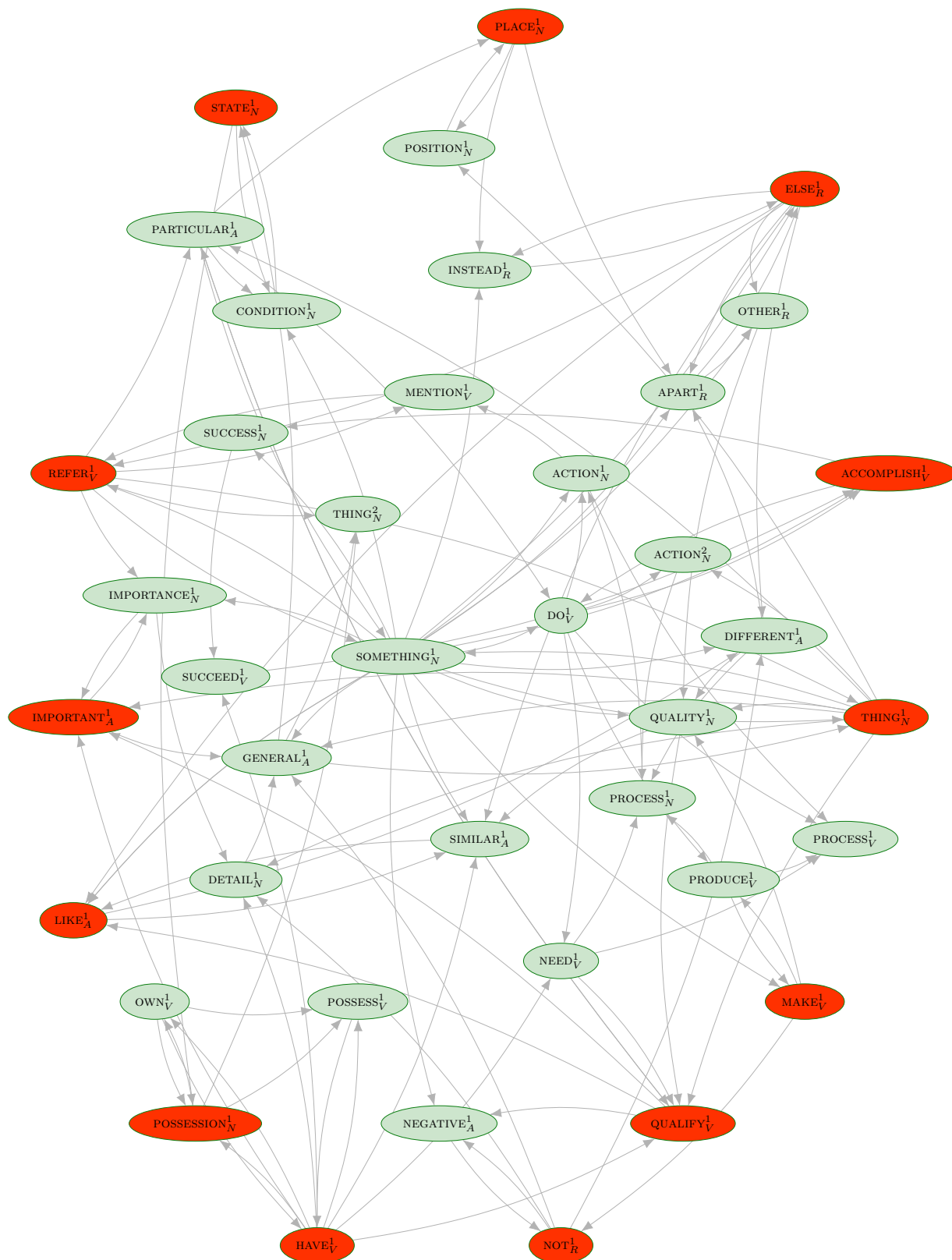


Figure 9. Graph associated to lexicon X_{large}
(Lexemes in the minimum feedback vertex set are marked in red).

The flow of Algorithm 1 is as follows:

- line 4:** Get next lexeme from strategy.
line 5: Make sure lexeme exists in lexicon.
line 7: Learn lexeme directly, at cost 1.
lines 8-10: Then learn by definition, at cost 0, all lexemes completely defined.
line 12: Repeat the preceding steps until both the lexicon and the strategy are empty.
line 13: The cost of the strategy is the sum of the learning cost for all lexemes learnt directly.

Algorithm 1

```

1: function PARTIALCOST( $S$  : strategy,  $X$  : lexicon)
   : (cost, lexicon)
2:    $cost \leftarrow 0$ 
3:   while  $S \neq \emptyset$  and  $X \neq \emptyset$  do
4:      $\ell \leftarrow S.POP()$ 
5:     if  $\ell \in X$  then
6:       Remove  $\ell$  from  $X$ 
7:        $cost \leftarrow cost + 1$ 
8:       while  $\exists \ell' \in X$  with  $\deg^-(\ell') = 0$  do
9:         Remove  $\ell'$  from  $X$ 
10:      end while
11:    end if
12:  end while
13:  return ( $cost, X$ )
14: end function

```

Algorithm 2: Dynamic Degree Learning Cost

The DYNAMICCOST algorithm calculates the cost to learn all the lexemes in a lexicon. At each iteration, it chooses the node having the maximum out-degree amongst the ones remaining in the graph. In other words, it directly learns the lexeme appearing in the largest number of definitions. The function DYNAMICCOST() accepts only one parameter:

- X , a lexicon.

The flow of Algorithm 2 is:

- line 3:** Get the lexeme that corresponds to the highest out-degree node from associated graph.
line 6: Learn lexeme directly, at cost 1.
lines 7-9: Learn by definition, at no cost, all lexemes completely defined.
line 10: Get next lexeme with highest out-degree.
line 11: Repeat preceding steps until the lexicon and the strategy are empty.
line 12: The strategy's cost is the sum of the learning cost for all lexemes learnt directly.

Algorithm 2

```

1: function DYNAMICCOST( $X$  : lexicon) : cost
2:    $cost \leftarrow 0$ 
3:    $\ell \leftarrow$  lexeme whose out-degree is highest
4:   while  $\ell \neq \emptyset$  do
5:     Remove  $\ell$  from  $X$ 
6:      $cost \leftarrow cost + 1$ 
7:     while  $\exists \ell' \in X$  with  $\deg^-(\ell') = 0$  do
8:       Remove  $\ell'$  from  $X$ 
9:     end while
10:     $\ell \leftarrow$  lexeme whose out-degree is highest
11:  end while
12:  return  $cost$ 
13: end function

```

Algorithm 3: Total Learning Cost

This algorithm calculates the total cost incurred for learning all lexemes in lexicon X using strategy S .

For exhaustive strategies, the total cost obtained with this algorithm is identical to the one obtained with Algorithm 1. For non-exhaustive strategies, the total cost obtained is the sum of strategy S 's cost, plus the cost incurred by applying to the remaining portion X' of the lexicon a fallback strategy. It is theoretically possible to devise different algorithms that could be used as a fallback strategy. In our case, we use the DYNAMICCOST dynamic out-degree computation method described in Algorithm 2.

The parameters for the function TOTALCOST() are:

- S , a learning strategy.
- X , a lexicon.

It returns as result:

- *total cost*: the total cost incurred by learning all lexemes in X .

Algorithm 3

```

1: function TOTALCOST( $S$  : strategy,  $X$  : lexicon)
   : cost
2:   ( $cost, X'$ )  $\leftarrow$  PARTIALCOST( $S, X$ )
3:   ( $total\ cost$ )  $\leftarrow cost +$  DYNAMICCOST( $X'$ )
4:   return  $total\ cost$ 
5: end function

```

Complexity Analysis:

The algorithms described in the previous section are very efficient and easy to implement. Here is an evaluation of their complexity:

Algorithm 1 : If we take as hypotheses that:

- G is the graph associated with lexicon X .
- n is the number of vertices in G .
- m is the number of arcs in G .
- At line 6, vertex removal is done in $\mathcal{O}(1)$.
- At line 8, we only look at the neighbors of the deleted vertices.

Then, the time complexity is $\mathcal{O}(n + m)$ and the space complexity is $\mathcal{O}(n)$.

Algorithm 2 : If we take as hypotheses that:

- (a) G is the graph associated with lexicon X .
- (b) n is the number of vertices in G .
- (c) m is the number of arcs in G .
- (d) At line 5, vertex removal is done in $\mathcal{O}(1)$.
- (e) At line 7, we only look at the neighbors of the deleted vertices.
- (f) At lines 3 and 10 the list of candidates is managed using a priority queue, in time $\mathcal{O}(\log n)$.
- (g) The priority queue is implemented using a heap

The total cost for line 3 is $\mathcal{O}(m \log n)$, since each vertex v is processed in $\mathcal{O}(\log n)$ at most $\mathcal{O}(\deg(v))$ times. The time complexity is therefore $\mathcal{O}(m \log n)$ and the space complexity $\mathcal{O}(n)$.

Algorithm 3 : The time complexity for Algorithm 3 is therefore $\mathcal{O}(m \log n)$ and the space complexity is $\mathcal{O}(n)$.

IV. DATA SETS

In this section, we present the data we used to study of the structure of the dictionaries. First of all, we describe the digital dictionaries from which the lexicons and associated graphs were built. All of them are works from professional lexicographers and are published in electronic format. Then, we look at the different learning strategies developed to “learn” the words in the lexicons. They are of two types:

- *psycholinguistic strategies*, built from specially labeled word lists, called psycholinguistic norms,
- *algorithmic strategies*, obtained by analyzing the structure of graphs associated with the lexicons.

A. Digital Dictionaries

As a basis for the analysis of lexicon’s structure, we used eight different monolingual English-language dictionaries developed by professional linguists. Most of them are available in digital or paper format, with the exception of Wordsmyth, available only on the web.

The Cambridge International Dictionary of English (CIDE) is an English-language dictionary developed for ESL – English as a Second Language – students [7]. The version we used comprises about 19,000 articles and 47,000 lexemes.

The Longman Dictionary of Contemporary English (LDOCE) is an advanced dictionary also for ESL students. It was first published in 1978 [6]. It includes about 29,000 articles and 70,000 lexemes.

These 2 dictionaries, CIDE and LDOCE, have a common feature [67], [68]. They are both “monolingual learners dictionaries” (MLD), that is dictionaries developed especially for the needs of second language students, in this case English [69, p. 739, Rundell]. Both of them were built from their own control vocabulary. In other words, all definitions use only words from a restricted vocabulary, making it easier for novice users to understand definitions. In both cases, the control vocabulary contains about 2000 lexemes.

The Merriam-Webster’s Collegiate Dictionary (MWC) is the largest dictionary we studied [21]. The 11th edition includes more than 250,000 lexemes, grouped into 70,000 articles.

WordNet (WN) is not a dictionary in the true sense of the word. It is rather a lexical database of the English-language [69, p. 665, Fellbaum]. The different lexemes are

regrouped into synonym sets or **synsets**. Each synset then refers to a “meaning” and to a gloss – definition –. Synsets are also connected to each other by different types of semantic relations, such as hyponymy, hyperonymy, etc. The version we used, WordNet 3.0, contains about 132,000 lexemes grouped into 57,000 synsets.

According to its authors, Wordsmyth is at the same time a dictionary and a thesaurus [41]. Unlike CIDE and LDOCE, it does not use a control vocabulary. However it offers, in addition to the definition of a given word, information about its synonyms, antonyms and similar words [41]. It is available in 4 versions:

- The “Wordsmyth Educational Dictionary-Thesaurus” (WEDT) is the most comprehensive, comprising 73,000 lexemes. It was first developed in the 1980s.
- The “Wordsmyth Illustrated Learner’s Dictionary” (WILD) is an illustrated dictionary for children. It includes 4,200 lexemes.
- The “Wordsmyth Learner’s Dictionary-Thesaurus” (WLDT) is an intermediate level dictionary. It comprises 6,000 lexemes.
- The “Wordsmyth Children’s Dictionary-Thesaurus” (WCDT) is a beginners dictionary. It contains 20,000 lexemes.

Using a sequence of pre-treatments, we transformed all these digital dictionaries into disambiguated and complete lexicons. To do this, we first extracted from each dictionary the words with the desired parts of the speech: noun, verb, adjective and adverb. In addition, we did not considered the compound lexical items in our analysis; they were ignored during the transformation of dictionaries into lexeme graphs. We then lemmatized and pos-tagged the lexemes in the definitions with the “Stanford POS-tagger” [70], again ignoring the stop words. Finally, we disambiguated the lexemes using the first sense heuristic.

Table V presents some basic statistical data for the 8 lexicons considered:

- The number of lexemes in each dictionary (Lexemes).
- The number of lemmas (Lemmas).
- The average polysemy, being the average number of lexemes per lemma.
- The number of lexemes used in the definitions (Lexemes used).
- The ratio of the number of lexemes used vs the total number of lexemes (Usage Ratio).

TABLE V. Basic statistical data on lexicons

Lexicon	Lexemes	Lemmas	Polysemy	Lexemes Used	Usage Ratio
WILD	4 244	3 081	1.377	2 995	0.972
WLDT	6 036	3 433	1.758	2 212	0.644
WCDT	20 128	9 303	2.164	6 597	0.709
CIDE	47 092	18 694	2.519	8 773	0.469
LDOCE	69 204	22 511	3.074	10 074	0.448
WEDT	73 091	28 986	2.522	18 197	0.628
WN	132 547	57 243	2.316	29 600	0.517
MWC	249 137	68 181	3.654	33 533	0.492

After building the graphs associated with the lexicons, we then analyzed their structure. Many measures can be applied

to networks or graphs. Among others, Batagelj *et al.*, identify a series of measures specifically aimed at dictionary graphs [71]. For our analysis, we selected the numbers that present a quick overview of the graphs. Table VI shows the results obtained from the graphs associated with the 8 lexicons:

- The number of vertices (Nodes).
- The number of arcs (Arcs).
- The number of strongly related components (SCCs).
- The number of lexemes in the largest SCC (<SCC).
- The diameter of the largest SCC (Diameter), being “[...] the largest number of vertices that must be traversed in order to travel from one vertex to another” [72,].
- The density of the graph (Density).
The density of a graph $G = (V, E)$ is the ratio of the number of arcs $|E|$ in G over the maximum number of arcs possible $= (|V| \cdot (|V| - 1)) / 2$ [73].
- The Characteristic Path Length (CPL) – the average length of the shortest paths – is calculated for a graph $G = (V, E)$ using the following formula [74]:

$$\sum_{u,v \in V} \frac{d(u,v)}{|V|(|V| - 1)}$$

TABLE VI. Associated graphs structural data

Lexicon	Nodes	Arcs	SCCs	<SCC	Diam.	Dens.	CPL
WILD	4 244	45 789	2 750	1 446	17	10.79	1.75
WLDT	6 036	28 623	5 088	858	25	4.74	1.10
WCDT	20 128	102 657	17 551	2 341	22	5.10	0.87
CIDE	47 092	334 888	45 306	1 702	16	7.11	0.21
LDOCE	69 204	415 052	67 224	1 770	16	6.00	0.16
WEDT	73 091	362 569	67 318	5 056	29	4.96	0.61
WN	132 547	694 067	124 589	7 079	30	5.24	0.50
MWC	249 137	1 155 085	239 478	8 842	29	4.64	0.31

B. Learning Strategies

There are a very large number of different strategies for learning all the words of a dictionary or lexicon. One could imagine trying them all. If a lexicon contains n lexemes, there are then $n!$ different ways to order them to specify a learning order. Except for trivial cases, it is obviously impossible to evaluate all those possibilities. We decided to restrict our study to two kinds of strategies:

- *Psycholinguistic Strategies*: These strategies are based on lists of words ordered according to psycholinguistic properties.
- *Algorithmic Strategies*: These strategies are built using algorithms from graph theory. Among these, one can distinguish the adapted strategies, built solely for a specific lexicon, and the global strategies based on normalized structural properties common to all lexicons.

Psycholinguistic Strategies:

Researchers interested in the cognitive aspects of language have long used standardized databases, called *psycholinguistic norms*, which group words according to their psycholinguistic properties [75]–[78]. For example, the MRC database lists 150,837 English-Language words, for which 26 different psycholinguistic properties are listed [78].

Among the recent psycholinguistic norms, we have selected five of them, made available by their authors as a supplement to their research work. They are lists of words based on psycholinguistic variables frequently used by researchers in language psychology: words *usage frequency*, *age of acquisition* and *degree of concreteness* [79]. The *usage frequency* measurement is probably the most commonly used norm for psycholinguistic research [66]. It is a measure of the rate of occurrence of words within a given corpus, normalized to 1 million. *Age of acquisition* is an estimation of the age at which children are presumed, on average, to have learned a word. As for the *concreteness*, it “[...] refers to the degree to which words refer to individuals, places and objects that can be seen, heard, touched, smelled or tasted” [80, cited by [75]].

Table VII presents the 5 data sources we used to construct our learning psycholinguistic strategies, versus the psycholinguistic variables from which they were derived. It goes without saying however that our analysis could easily be extended to other databases or other variables, depending on data availability.

TABLE VII. Psycholinguistic Variables and Learning Strategies

Variable	Strategy	Source	# Words
Usage frequency	FREQ _{Brysbaert}	[66]	74 000
Usage frequency	FREQ _{NGSL+}	[65], [81], [82], [83]	6 600
Age of acquisition	AOA _{Brysbaert}	[84]	31 000
Age of acquisition	AOA _{Childes}	[85]	13 000
Concreteness	CONC _{Brysbaert}	[86]	37 000

To build our learning strategies, we first lemmatized and disambiguated the words from the databases in order to transform them into lexemes, and then ordered them according to the psycholinguistic variable considered. For example, for a strategy based on the age of acquisition, the first lexeme proposed by the strategy corresponds to the word that the authors consider to be learned the earliest in the development of the child. Then the second lexeme suggested corresponds to the second word learned and so on until we get to the lexeme estimated to be learned the latest.

An additional alignment step between lexicons and strategies is required. Since the psycholinguistic data used to construct the strategies come from heterogeneous sources, the lexemes they contain do not necessarily match with the lexicons. When a lexeme proposed by a strategy does not appear in a lexicon, we choose to simply ignore it. In particular, we do not measure the degree alignment of psycholinguistic strategies with lexicons, that is, the size of the intersections between the strategies and the lexicons. This is one of the limitations of our analysis. If we were to tackle it in the future, this could possibly allow a more refined assessment of the quality of the strategies.

Let us now look at how the different learning strategies were developed.

The first strategy in table VII, FREQ_{Brysbaert} is derived from the norm described in [66]. The authors assembled it from SUBTLEX_{US}, a corpus of film subtitles in American English. It includes 74,000 unlemmatized words.

The FREQ_{NGSL+} strategy comes from lists of words used to learn English as a second language. Although word lists

are not based solely on psycholinguistic criteria, they are still an important domain of research since the works of Ogden [59] and West [61]. For this paper, we selected the “New General Service List” (NGSL) from Browne, Culligan and Phillips [65]. This is an improved version of West’s original list, containing 2,800 words selected from the Cambridge English Corpus (CEC). To enable the NGSL to be more easily compared to other psycholinguistic strategies, we developed an augmented version: the NGSL+. The latter is obtained by concatenating to the NGSL three other lists of complementary words developed by the authors of the NGSL from specialized corpora:

- The New Academic Word List (NAWL) is constructed from a body of academic texts [81]. It contains 963 words.
- The Business Service List (BSL) is a list of 1700 words related to business and commerce [82].
- The “TOEIC Service List” (TSL) is intended for students wishing to attain the “Test of English for International Communication” (TOEIC) certification. It is a list of 1200 words that complement the NGSL [83].

To build the AOA_{Brysbaert} strategy, we used the norm based on the age acquisition norm from Kuperman *et al.* [84]. Since it is not possible to get this information directly from the children themselves, the most frequently used method is to interview adults and ask them to assess the age at which they have learned certain words. For their research, Kuperman, Stadthagen-Gonzalez and Brysbaert used a crowdsourcing technique based on the Amazon Mechanical Turk. Adult participants were asked to estimate how old they were when they learned the words from a list. From their responses, the authors constructed a list of 31,000 words tagged with their estimated age of acquisition, ranging from 1 to 21 years old.

The other age-based acquisition strategy, AOA_{Chilides}, uses data from another source: the project “Child Language Data Exchange System” (CHILDES) [85]. In this case, a different method was used to collect the data. The age of acquisition was estimated from recorded conversations of children aged 1 to 11 years. The resulting list, noisier than the previous one, contains 13,000 words.

For the CONC_{Brysbaert} strategy, we used Brysbaert, Wariner and Kuperman’s norm [86]. As with their study on the age of acquisition, the authors used crowdsourcing to recruit participants. The adults chosen had to classify words on a concreteness scale ranging from 1 to 5, 1 being completely abstract and 5 corresponding to the most concrete words. For example, the concrete words *banana*, *apple* and *baby* are of degree 5, while *belief* and *although* are respectively of degree 1.19 and 1.07. The list thus created contains 37,000 words.

Algorithmic Strategies:

Algorithmic strategies are lists of lexemes derived from the structural properties of graphs, which means that lexemes are ordered according to the results of graph theory algorithms.

Table VIII summarizes the algorithmic strategies we have experimented with. It should be noted that all these strategies directly use the COST or DYNAMICCOST algorithms without resorting to a fallback strategy. In contrast to psycholinguistic strategies, the techniques used ensure that lexicons are fully “learned” when the algorithms terminate.

With the first 3 strategies, MFVS_{<lex>}, DD_{<lex>} and SD_{<lex>}, we get as many different strategies as lexicons,

TABLE VIII. Algorithmic learning strategies

Strategy	Property	Algorithm	Number
MFVS _{<lex>}	Min. Grounding Set	COST	8 (1 per lexicon)
DD _{<lex>}	Dynamic Degree	DYNAMICCOST	8 (1 per lexicon)
SD _{<lex>}	Static Degree	COST	8 (1 per lexicon)
MFVS _{mixed}	Min. Grounding Set	COST	1
DD _{mixed}	Dynamic Degree	COST	1
SD _{mixed}	Static Degree	COST	1

each one of them being adapted to a specific lexicon. Here the <lex> index represents the lexicon. For instance, SD_{LDOCE} corresponds to the static degree strategy for the LDOCE lexicon.

The MFVS_{<lex>} strategies are assembled individually for each lexicon <lex> from the minimum grounding set calculated with the method described in definition 8. Although the problem of calculating a MFVS is NP-hard in general, it was still possible to obtain an optimal solution for 6 of the 8 lexicons and a good approximation for the 2 others. In this specific case, the order of the lexemes in the strategy is not considered.

With the DD_{<lex>} Dynamic Degree strategies, the next lexeme to be learned is not chosen from a predetermined list. As described in Algorithm 2, it is calculated dynamically at each step by selecting the vertex whose out-degree is the highest. Since “learned” lexemes are systematically removed at each step, it is equivalent to selecting each time the lexeme that appears in the greatest number of definitions.

For the SD_{<lex>} Static Degree strategies, the next lexeme to learn comes from a list containing all the lexicons of the lexicon. The lexemes are ordered in descending order of the out-degree of their corresponding vertices. Unlike the DD_{<lex>} strategies, the degree of vertices is computed statically when the graph is initially built. Thus, one begins to learn the lexemes from the one that is used in most definitions, going to the least used.

In order to evaluate whether the use of strategies uniquely built for each lexicon could distort the results, we also developed global strategies, based on structural data common to all the lexicons. Those strategies, called *mixed strategies*, are assembled by merging into one global list all the lexemes coming from the strategies adapted to each lexicon. For example, the lexemes from the 8 DD_{<lex>} strategies are merged to form the DD_{mixed} list. It is built by randomly choosing one of the 8 lexicons, and then selecting the next lexeme from the corresponding strategy. If the lexeme is already in the global list, it is ignored. We then repeat this process until all the lists are exhausted. For example, the DD_{mixed} strategy was built by concatenating the lexemes in the order shown in Table IX.

V. RESULTS AND DISCUSSION

In this section, we present the results obtained during our experiments. First, we explain the different measures collected during the execution of the algorithms on the lexicons. We then show comparative results for the various learning strategies and the 8 lexicons analyzed. We conclude the section with a discussion of the results.

TABLE IX. Mixed learning strategies

Number	Lexeme	Origin
1.	BE;V	DD _{WILD}
2.	HAVE;V	DD _{WN}
3.	PERSON;N	DD _{WLDT}
4.	USE;N	DD _{WN}
...
5990.	DEALFISH;N	DD _{MWC}
5991.	PHENYTOIN;N	DD _{MWC}

A. Measurements

To allow for the combinations of strategies and lexicons to be compared, different performance indicators were recorded during the tests.

Detailed Learning Measurements:

When learning a lexicon with a strategy, a series of values is recorded each time a lexeme is learned directly. This makes it possible to evaluate the pace of the learning process. Table X shows an overview of the data recorded during one learning cycle of the MWC lexicon using the AOA_{Brybaert} strategy.

TABLE X. Learning progress (partial)

Cost	Nodes	Arcs	Degree	Lexeme	Fallb.
1	249 056	1 152 896	2	mama;n	0
2	249 054	1 152 894	2	mom;n	0
3	249 053	1 152 892	8	potty;n	0
4	249 051	1 152 884	17	yes;n	0
5	249 047	1 152 867	1 522	water;n	0
6	249 039	1 151 337	130	wet;a	0
7	249 037	1 151 208	33	spoon;n	0
8	249 036	1 151 175	51	nap;n	0
9	249 030	1 151 121	2	daddy;n	0
10	249 028	1 151 119	18	hug;n	0
11	249 026	1 151 101	212	shoe;n	0
10	249 028	1 151 119	18	hug;n	0
11	249 026	1 151 101	212	shoe;n	0
...
10113	14	14	1	kakemono;n	484
10114	12	12	1	stilbestrol;n	485
10115	10	10	1	ciphertext;n	486
10116	8	8	1	banderilla;n	487
10117	6	6	1	amphitryon;n	488
10118	4	4	1	mannose;n	489
10119	2	2	1	phenytoin;n	490

We can see the following measures:

Cost: Number of lexemes learned directly since the beginning of the cycle

Nodes: Number of vertices remaining in the graph (before learning the lexeme)

Arcs: Number of arcs remaining in the graph (before learning the lexeme)

Degree: Out-degree of the lexeme

Lexeme: Lexeme learned

Fallb.: Cumulative cost of the fallback strategy

Global Performance Measurements:

At the end of a learning cycle, global performance indicators are also recorded. Table XI shows, for each combination of lexicons and learning strategies, their performance in terms of cost, efficiency, percentage of words learned directly, and coverage (if applicable).

The counters shown are:

Cost: Indicates the total learning cost for the strategy, i.e., the total number of lexemes that had to be learned directly in order to successfully learn the full lexicon (see algorithms 1 and 2). For example, the cost for the DD_{WILD} strategy and the WILD lexicon is 574. This represents the total number of lexemes that had to be learned directly.

Efficiency: Efficiency is the ratio of the total number of lexemes over the cost of learning. We see that the WILD lexicon contains 4,244 lexemes and that 1,260 lexemes had to be learned directly with the FREQ_{NGSL+} strategy. The efficiency of the FREQ_{NGSL+} for the WILD lexicon is therefore 4,244/1,260 or 3.37. We can interpret this measure as the average number of lexemes that can be learned by definition for each lexeme learned directly. In other words, we learn on average 2.37 additional lexemes by definition every time we learn a lexeme directly.

Pct: Percentage of words learned directly for a given strategy and lexicon. This corresponds to the proportion of the number of lexemes learned directly, relative to the total number of lexemes in the lexicon. For example, for the WEDT lexicon and the FREQ_{NGSL+} strategy, the percentage of lexemes learned directly is 3 238 over 73 091 or 4.43%.

Coverage: Valid for non-exhaustive strategies only, this number measures the efficiency of the strategy, as a percentage of the total cost, vs the fallback strategy. For example, for the WCDT lexicon and the FREQ_{NGSL+} strategy, the coverage is 82.9%. This means that out of a total learning cost of 1,354, 1,122 lexemes, or 82.9%, were learned with the FREQ_{NGSL+} strategy. The remaining 232 (17.1%) were learned with DD, our fallback strategy. On the other hand, for this same WCDT lexicon and the FREQ_{Brybaert} strategy, the coverage reaches 99.8%.

It turns out that, unsurprisingly, the most efficient strategies are those that take advantage of the minimal grounding set: the MFVS_{<lex>}. Coming right after, the strategies optimized according to the vertices out-degree - DD_{<lex>} and DS_{<lex>} - are also very efficient. We also remark that for some lexicons - MWC, WN, WEDT, WCDT - the FREQ_{NGSL+} and AOA_{Chilides} strategies have a low coverage rate of less than 90%.

B. Discussion

Global Performance Measurements:

The Figures in this section (Best viewed in colors) compare different aspects of the learning process for the 8 lexicons studied. Each of the sub-figures is produced using the detailed performance measurements recorded while lexemes are being learned.

The first series of graphs in Figure 10 compare the learning rate of algorithmic strategies versus psycholinguistic strategies.

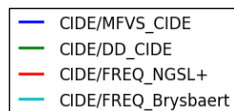
To facilitate comprehension, let us examine Figure 10a for the CIDE lexicon. It shows the learning rate for the algorithmic strategies MFVS_{CIDE} and DD_{CIDE} in comparison with the FREQ_{NGSL+} and FREQ_{Brybaert} psycholinguistic strategies.

TABLE XI. Cost, efficiency, percentage and coverage

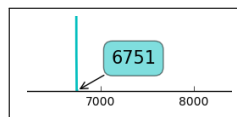
Strategy	Measure # Lex	CIDE	LDOCE	MWC	WN	WEDT	WCDT	WLDT	WILD
		47 092	69 204	249 137	132 547	73 091	20 128	6 036	4 244
MFVS	Cost	349	484	1 544	1 251	1 365	570	231	340
	Eff.	134.93	142.98	161.36	105.95	53.55	35.31	26.13	12.48
	Pct	0,74%	0,70%	0,62%	0,94%	1,87%	2,83%	3,83%	8,01%
	Cov.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.
DD	Cost	684	843	3 095	2 566	2 389	897	394	574
	Eff.	68.85	82.09	80.50	51.66	30.59	22.44	15.32	7.39
	Pct	1,45%	1,22%	1,24%	1,94%	3,27%	4,46%	6,53%	13,52%
	Cov.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.
DS	Cost	687	838	3 081	2 558	2 386	899	394	577
	Eff.	68.55	82.58	80.86	51.82	30.63	22.39	15.32	7.36
	Pct	1,46%	1,21%	1,24%	1,93%	3,26%	4,47%	6,53%	13,60%
	Cov.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.
MFVS _{mixed}	Cost	704	966	3 077	2 835	2 348	957	398	612
	Eff.	66.85	71.64	80.96	46.75	31.13	21.03	15.17	6.93
	Pct	1,49%	1,40%	1,24%	2,14%	3,21%	4,75%	6,59%	14,42%
	Cov.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.
DD _{mixed}	Cost	768	963	3 466	3 002	2 574	987	448	645
	Eff.	61.32	71.82	71.88	44.15	28.39	20.39	13.47	6.57
	Pct	1,63%	1,39%	1,39%	2,26%	3,52%	4,90%	7,42%	15,20%
	Cov.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.
DS _{mixed}	Cost	793	988	3 776	3 021	2 721	1 024	454	678
	Eff.	59.32	70.00	65.98	43.87	26.86	19.65	13.30	6.25
	Pct	1,68%	1,43%	1,52%	2,28%	3,72%	5,09%	7,52%	15,98%
	Cov.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.	s. o.
FREQ _{NGSL+}	Cost	2 813	1 954	5 008	4 127	3 238	1 354	712	1 260
	Eff.	16.74	35.42	49.75	32.12	22.57	14.87	8.48	3.37
	Pct	5,97%	2,82%	2,01%	3,11%	4,43%	6,73%	11,80%	29,69%
	Cov.	97.0%	90.4%	71.2%	73.4%	67.7%	82.9%	97.9%	92.8%
FREQ _{Brysb}	Cost	6 751	2 170	8 217	7 204	6 555	1 999	960	1 193
	Eff.	6.98	31.89	30.32	18.40	11.15	10.07	6.29	3.56
	Pct	14,34%	3,14%	3,30%	5,44%	8,97%	9,93%	15,90%	28,11%
	Cov.	99.9%	99.3%	96.1%	94.8%	98.7%	99.8%	99.7%	99.6%
AOA _{Chil}	Cost	4 971	5 010	7 729	7 284	5 586	3 409	1 585	2 016
	Eff.	9.47	13.81	32.23	18.20	13.08	5.90	3.81	2.11
	Pct	10,56%	7,24%	3,10%	5,50%	7,64%	16,94%	26,26%	47,50%
	Cov.	99.4%	97.7%	82.9%	86.3%	84.3%	97.3%	99.7%	98.3%
AOA _{Brysb}	Cost	7 105	4 851	10 119	10 340	8 278	2 950	1 284	1 430
	Eff.	6.63	14.27	24.62	12.82	8.83	6.82	4.70	2.97
	Pct	15,09%	7,01%	4,06%	7,80%	11,33%	14,66%	21,27%	33,69%
	Cov.	99.6%	99.2%	95.2%	94.0%	96.7%	97.6%	99.5%	95.7%
CONC _{Brysb}	Cost	8 900	11 669	16 580	17 037	12 792	6 042	2 373	2 477
	Eff.	5.29	5.93	15.03	7.78	5.71	3.33	2.54	1.71
	Pct	18,90%	16,86%	6,65%	12,85%	17,50%	30,02%	39,31%	58,36%
	Cov.	99.7%	99.6%	96.4%	96.0%	97.5%	98.9%	99.7%	97.6%

We can see:

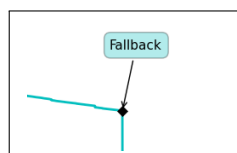
- The curves illustrating the learning rate, identified by a different color for each lexicon:



- A tile of the same color as the curve, showing for each strategy the total cost incurred:



- Another tile showing, for each non-exhaustive strategy, the point where it was required to resort to the fallback strategy:



For the CIDE lexicon (Figure 10a) as well as for all the lexicons in Figure 10, we see that the MFVS strategy is the most efficient one. This confirms the hypothesis that learning the minimal grounding set lexemes allows to quickly break the definition loops.

The graphs in Figure 11 show the learning rate for dynamic degree strategies versus those based on the static degree. We note that these two algorithmic strategies, DD <LEX> and SD <LEX> give in practice equivalent results.

The graphs in Figure 12 compare the learning rate for the algorithmic strategy DD<LEX> versus the psycholinguistic strategies FREQ_{NGSL+}, FREQ_{Brysbart}, AOA_{Brysbart} and CONC_{Brysbart}. We see that psycholinguistic strategies are much less effective in breaking the definition loops. Since the lexemes order is decided according to psycholinguistic criteria, many lexemes are learned directly and increase the cost of the strategy, whereas they could have been learned by definition - at zero cost - later in the learning cycle. Among the psycholinguistic strategies, the two frequency-based strategies, FREQ_{NGSL+} and FREQ_{Brysbart}, are the most effective, whereas the CONC_{Brysbart} strategy is clearly less efficient. Intuitively, we see that it is not possible to succeed in learning all the

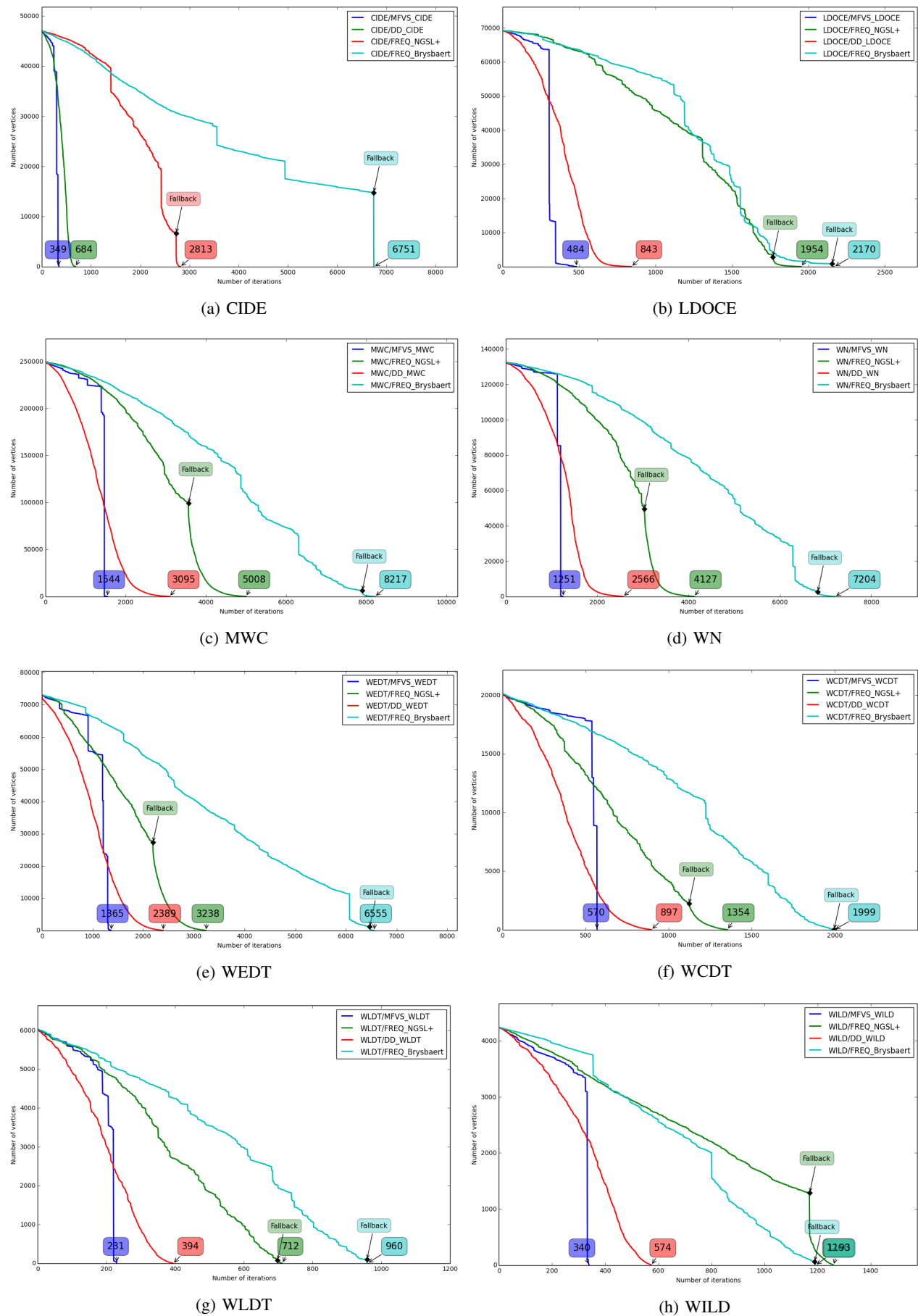


Figure 10. Learning: Algorithmic vs Psycholinguistic Strategies

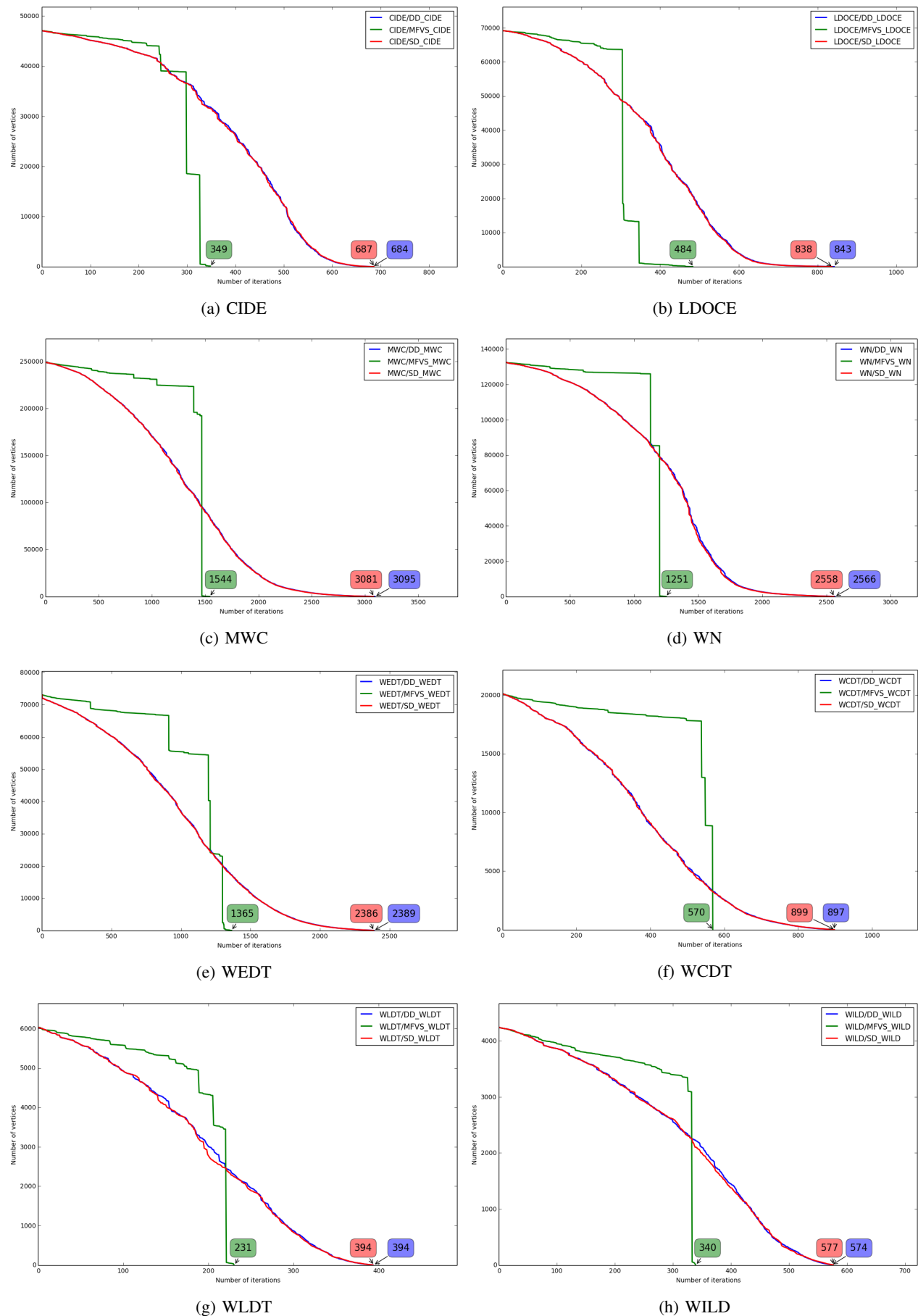


Figure 11. Learning: Dynamic vs Static Strategies

words of a dictionary by using only concrete words. One must combine two kinds of words, abstract and concrete, to build definitions that properly translate the lexical meaning.

Figure 13 compares the learning rate of age-based strategies. At first glance, AOA_{Childes} seems to offer a better efficiency than AOA_{Brybaert} . However, since OA_{Childes} contains far fewer lexemes than AOA_{Brybaert} , its coverage is lower. The fallback point is reached very soon, before having learned 40% of the lexemes. In this case, the use of a fallback strategy makes direct comparison between AOA_{Childes} and AOA_{Brybaert} difficult. That being said, although the AOA_{Childes} and AOA_{Brybaert} strategies are not the most successful ones, they show that in most cases it is sufficient to know less than 15% of the lexemes to learn the rest by definition.

The graphs in Figure 14 compare the mixed algorithms learning rate against the most effective psycholinguistic strategies. Since they are optimized to use as few lexemes as possible, algorithmic strategies are clearly more efficient. Lexemes ordering is the key factor making one strategy more efficient than the other. We notice large differences in this regard when comparing strategies based on the same psycholinguistic criteria. For example, the word dog appears at 485th rank in AOA_{Childes} , while it is ranked 25th in AOA_{Brybaert} .

Efficiency:

Table XI presents overall performance measurements for learning strategies. Figure 15 (best viewed in colors) plots efficiency for each evaluated lexicon and strategy. Each lexicon is each represented by a color coded curve. The strategies are shown on the X axis, from left to right in descending order of efficiency.

We can distinguish 3 different groups of strategies:

- 1) The 1st group comprises only one strategy: the algorithmic $MFVS_{\langle \text{lex} \rangle}$. For every lexicon, it is clearly the most efficient.
- 2) The second group gathers the other graph algorithmic strategies. $DD_{\langle \text{lex} \rangle}$ and $SD_{\langle \text{lex} \rangle}$ are uniquely optimized for each lexicon, while $MFVS_{\text{mixte}}$, DD_{mixte} and SD_{mixte} are global strategies common to all lexicons. They are less efficient than $MFVS_{\langle \text{lex} \rangle}$, but still very good.
- 3) Finally, the third group brings together the psycholinguistic strategies $FREQ_{\text{NGSL+}}$, $FREQ_{\text{Brybaert}}$, AOA_{Childes} , AOA_{Brybaert} and $CONC_{\text{Brybaert}}$. Their performance is clearly inferior compared to algorithmic strategies.

In summary, the uniquely optimized strategies, $MFVS_{\langle \text{lex} \rangle}$, $DD_{\langle \text{lex} \rangle}$ and $SD_{\langle \text{lex} \rangle}$, are the most efficient ones. As for the question of whether it is possible to develop “general” strategies as efficient as “lexicon specific” strategies, the mixed strategies show that this is possible. The 3 mixed strategies, $MFVS_{\text{mixte}}$, DD_{mixte} and SD_{mixte} are almost as efficient as the “lexicon specific” strategies. For each lexicon, they perform much better than strategies based on psycholinguistic variables.

VI. CONCLUDING REMARKS

By definition, a traditional dictionary is a closed world. According to Amsler, “[...] the dictionary is a closed system, i.e., words used in definitions are defined elsewhere in the dictionary” [87, p. vii]. It is therefore possible to build a graph structure from the words of a dictionary and the definitions

that link them together. In this article, our goal was to use graph theory and algorithms to study these dictionary graph structure.

Although the terms dictionary and word may seem a priori clear and unambiguous, their imprecision makes them unsuited for rigorous mathematical analysis. We decided to replace them in our discussion by more precise terms: lexicons and lexemes. We also established a linguistic terminology allowing to formally define the notions of lexicon and associated graph.

In order to explore the structure of lexicons, we have shown the interest of using a formal word learning process as an analytical tool. We considered that a word - a lexeme - can be learned in two ways: by definition, when all the lexemes in its definition are already known, and by direct learning, when one needs to invest significant effort to ground it through some sensorimotor perception. We also described our learning model, as well as related strategies and algorithms aiming to minimize the effort and cost required to learn all the lexicons of a lexicon.

Subsequently, we described the source data used to carry out our analyzes: monolingual digital dictionaries and psycholinguistic norms.

Finally we exposed our results in two different ways:

- in terms of learning rate, which is a measure of how quickly a strategy progresses toward its goal of learning all lexemes;
- in terms of efficiency, being the ratio between the number of lexemes learned by definition and the number of lexemes learned directly.

Our analysis confirmed the results of other researches ([15], [16], [55]). Circular relationships between words play a key role in the organization and structure of dictionaries.

If we consider a dictionary from the strict point of view of its utility for the reader, the definition of a word will be relevant insofar as the latter already knows all the words that make up this definition, or at least enough words to understand the intended meaning.

“The usefulness of a dictionary definition depends on its ability to explain a meaning using words the reader already knows” [88].

If this is not the case, the reader must look for unknown words. And in all dictionaries, there are necessarily many circular definitions:

“In a typical dictionary, more than a quarter of all definitions are written using words whose definitions ultimately refer back to the word being defined” [88]

A reader who does not know enough of the language will inevitably encounter intractable definition loops. Our analysis has shown that the most efficient learning strategies are those that break those definition loops as quickly as possible. In this regard, those who use the minimum grounding set - feedback vertex set of the associated graph - work best. However, the problem of finding a feedback vertex set is NP-hard. Even using advanced approximation techniques, this remains a complex calculation.

Our results show that alternative strategies, built using simple graph properties, can also be very efficient. For example, with a strategy ordering lexemes according to the out-degree

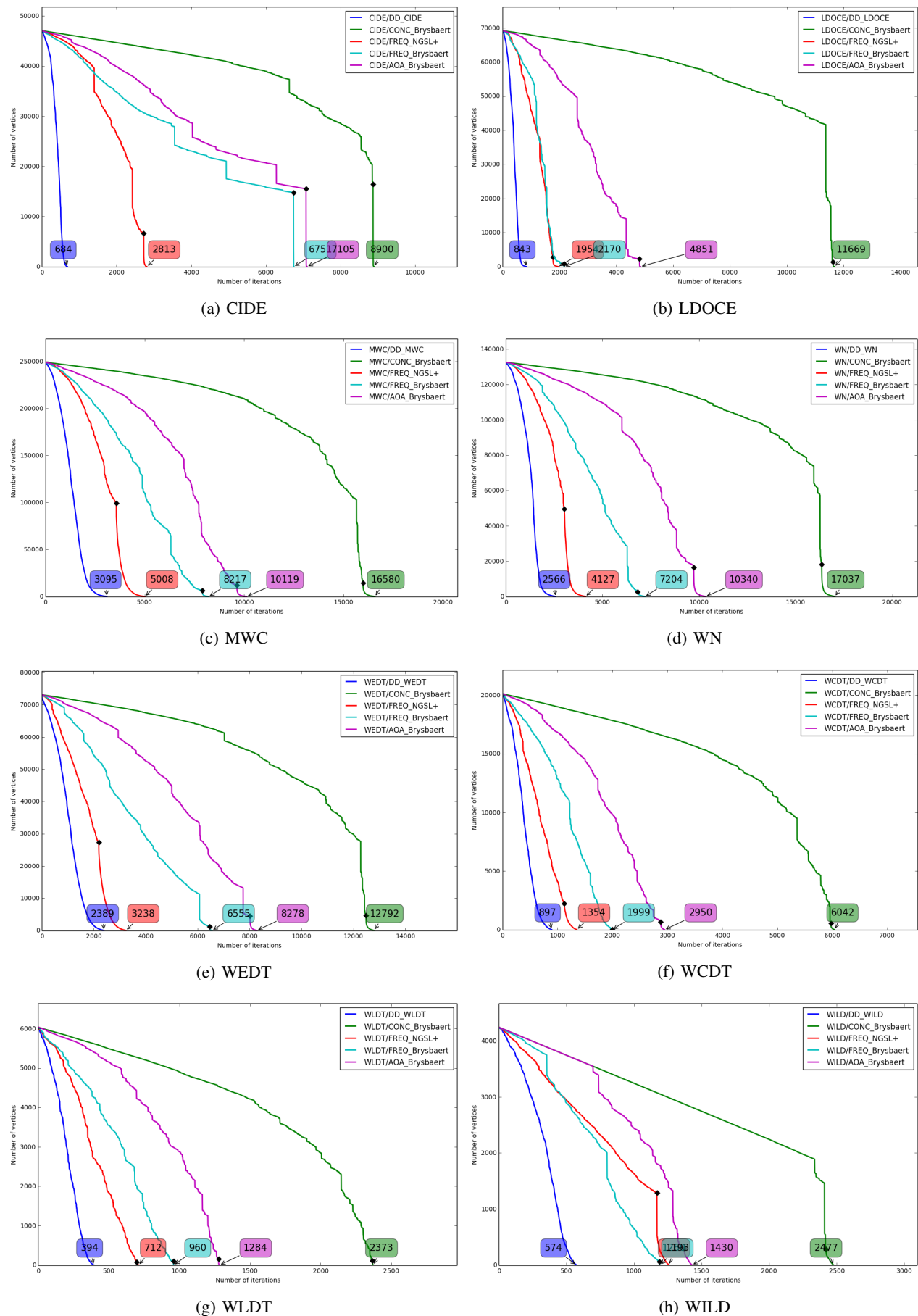


Figure 12. Learning: Frequency

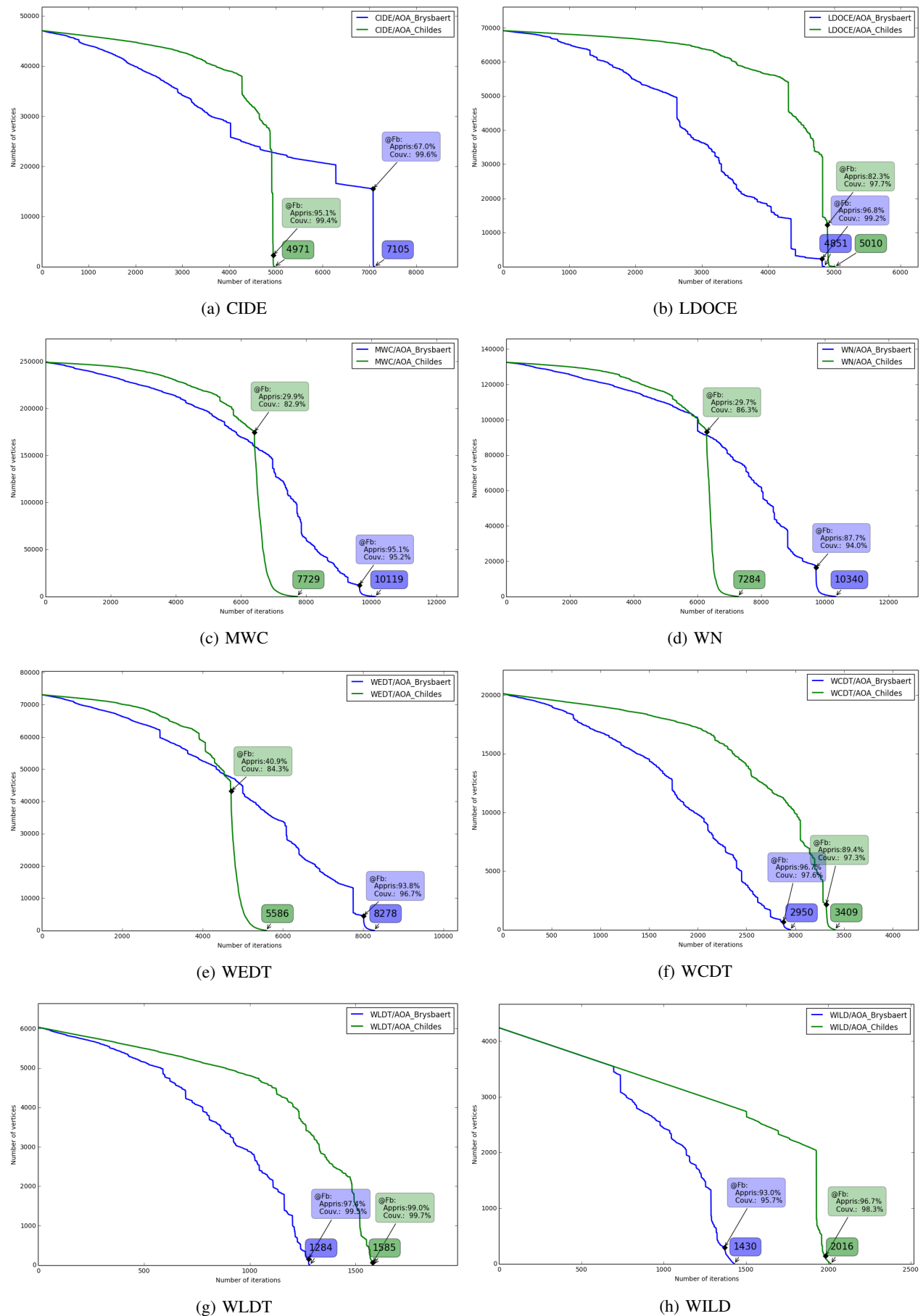


Figure 13. Learning: AOA based Strategies

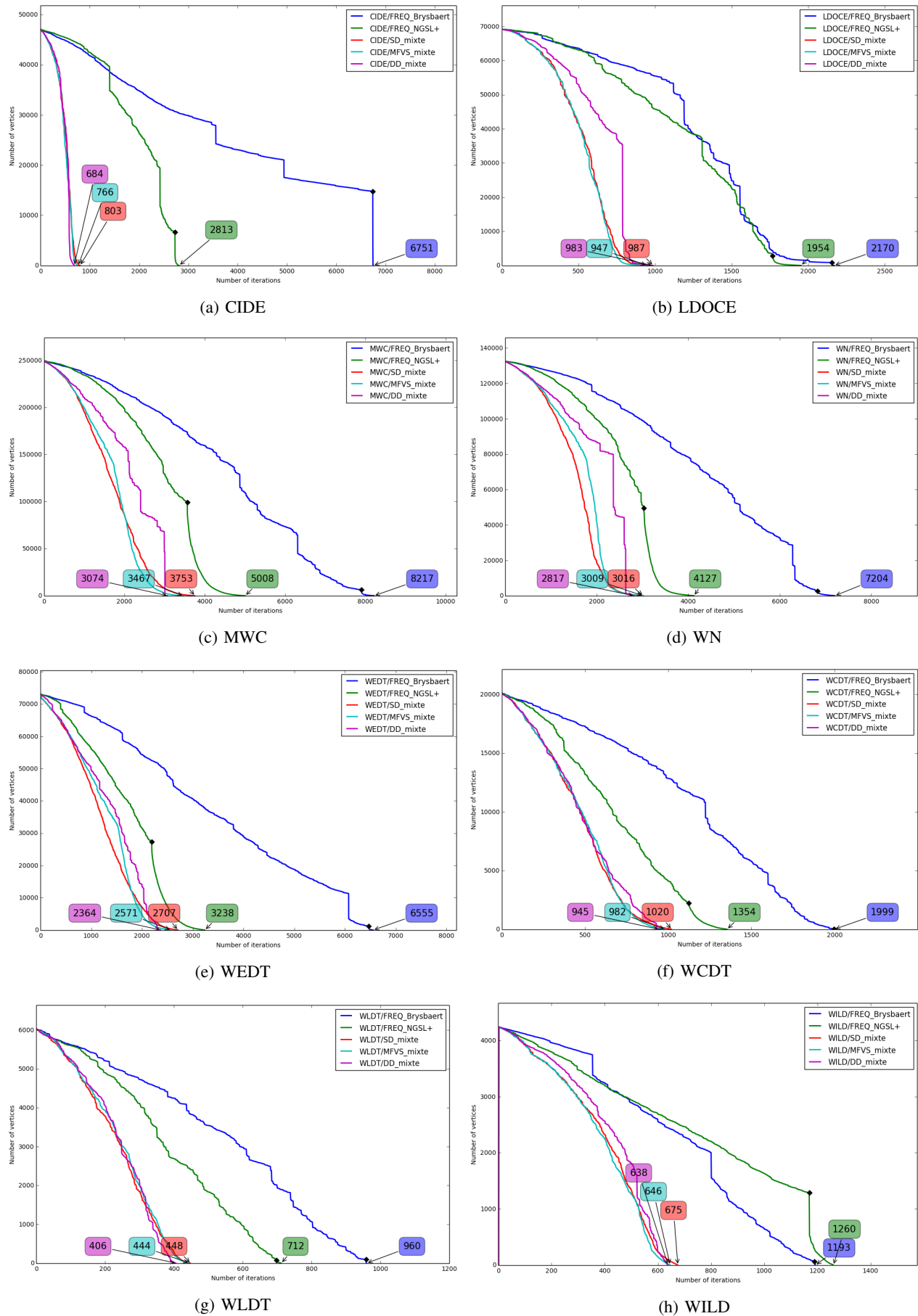


Figure 14. Learning: Optimized Strategies vs Mixed

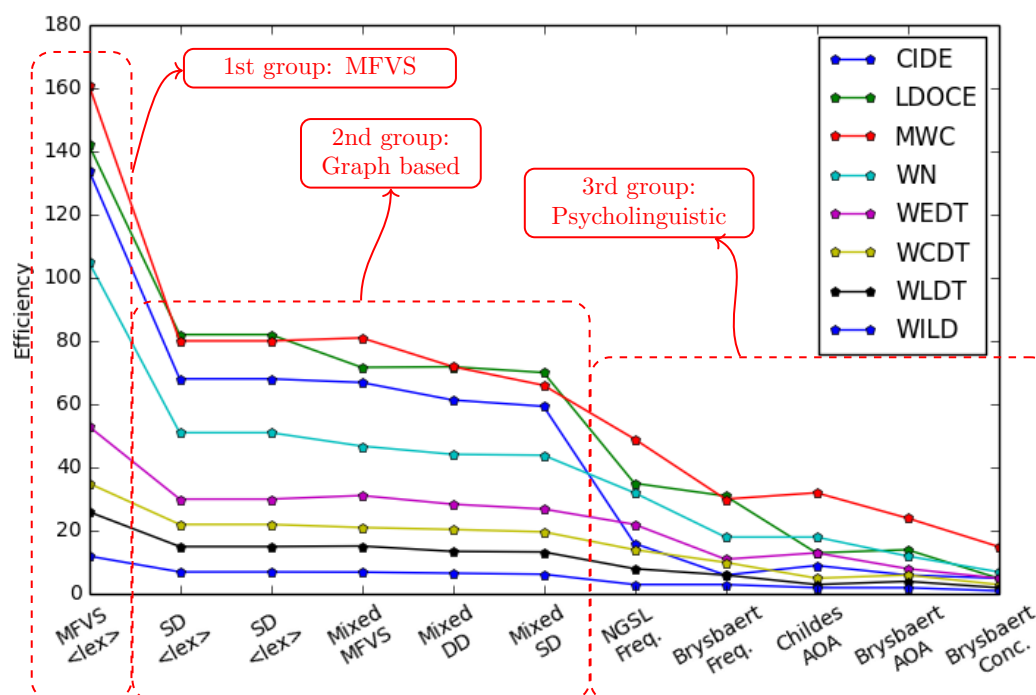


Figure 15. Lexicons: Efficiency vs Strategies

of their corresponding nodes, one can learn rapidly all the lexemes of a lexicon. In practical terms, this corresponds to a list of words ordered according to the number of times words appear in the definition of other words. Such a method by far outperforms psycholinguistic strategies.

In addition to their use as a dictionary analysis tool, the algorithmic strategies examined present additional advantages. To our knowledge, they represent a new approach for the development of word lists similar to those used in language teaching.

The word lists used by ESL teachers have traditionally been corpus-based, that is mainly built according to words frequency in a corpus. As an alternative, we propose a simple algorithmic strategy, based on the out-degree of vertices.

Although it is not possible to claim that the value of a word list is limited to its “efficiency”, we believe that this new approach could be used with profit, especially in cases where it is not possible to use existing word lists. In this case, or in the absence of an established corpus, the use of a digital lexicon or specialized dictionary would allow to establish easily a list of the relevant words or concepts, as well as the order in which they should be learned.

Future perspectives

A few words about the many research tracks left unexplored will conclude this article.

One might first think of extending the field of experimentation to new sources of data. Whether using new dictionaries, different digital lexicons, other algorithms for graph analysis, or new psycholinguistic norms, the possibilities are numerous. Similarly, one could explore dictionaries in fields such as medicine, mathematics, music or other specialized domains.

In addition, although resources in this area are often quite difficult to obtain, other languages would definitely offer rewarding research avenues. The analysis of monolingual dictionaries for languages other than English, or even of bilingual dictionaries, would for sure present many challenges.

Finally, the use of more advanced techniques to lexically disambiguate the definitions would offer a significant improvement to our methodology. Although the first sense heuristic usually gives satisfactory results and constitutes a strong baseline, newer techniques using neural networks and deep learning would certainly be worthwhile to explore. Improved word sense disambiguation as well as handling of compound lexical items would allow to build associated graphs more representative of underlying dictionaries and lexicons.

REFERENCES

- [1] J.-M. Poulin, A. B. Massé, and A. Fonseca, “Strategies for learning lexemes efficiently: A graph-based approach,” in *COGNITIVE 2018: The Tenth International Conference on Advanced Cognitive Technologies and Applications*. ThinkMind, 2018, pp. 18–23.
- [2] J.-C. Boulanger, *Les inventeurs de dictionnaires [ressource électronique] : De l'eduba des scribes mésopotamiens au scriptorium des moines médiévaux*. Canadian electronic library., ser. Collection Regards sur la traduction. Ottawa, Ont.]: Presses de l'Université d'Ottawa, 2003.
- [3] Merriam-Webster. (2018) Merriam-webster. [Online]. Available: <https://www.merriam-webster.com>
- [4] Collins. (2018) Collins dictionary. [Online]. Available: <https://www.merriam-webster.com>
- [5] G. Clark, “Recursion through dictionary definition space: Concrete versus abstract words,” *On WWW at* <http://www.ecs.soton.ac.uk/Åharnad/Temp/concreteabstract.pdf>. Accessed, vol. 23, no. 06, 2003.

- [6] P. Procter, *Longman Dictionary of Contemporary English (LDOCE)*. Essex, UK: Longman Group Ltd., 1978.
- [7] —, *Cambridge International Dictionary of English (CIDE)*. Cambridge University Press, 1995.
- [8] M. Steyvers and J. B. Tenenbaum, "The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth," *Cognitive science*, vol. 29, no. 1, pp. 41–78, 2005.
- [9] C. Fellbaum, Ed., *WordNet An Electronic Lexical Database*. Cambridge, MA ; London: The MIT Press, May 1998.
- [10] P. M. Roget, *Roget's Thesaurus of English Words and Phrases...* TY Crowell Company, 1911.
- [11] A. Blondin Massé, G. Chicoisne, Y. Gargouri, S. Harnad, O. Picard, and O. Marcotte, "How is meaning grounded in dictionary definitions?" in *Proceedings of the 3rd Textgraphs Workshop on Graph-Based Algorithms for Natural Language Processing*. Association for Computational Linguistics, 2008, pp. 17–24.
- [12] O. Picard, A. Blondin-Massé, S. Harnad, O. Marcotte, G. Chicoisne, and Y. Gargouri, "Hierarchies in dictionary definition space," *arXiv preprint arXiv:0911.5703*, 2009.
- [13] O. Picard, A. Blondin Massé, and S. Harnad, "Learning word meaning from dictionary definitions: Sensorimotor induction precedes verbal instruction," 2010.
- [14] O. Picard, M. Lord, A. Blondin-Massé, O. Marcotte, M. Lopes, and S. Harnad, "Hidden structure and function in the lexicon," *arXiv preprint arXiv:1308.2428*, 2013.
- [15] P. Vincent-Lamarre, A. B. Massé, M. Lopes, M. Lord, O. Marcotte, and S. Harnad, "The latent structure of dictionaries," *Topics in cognitive science*, vol. 8, no. 3, pp. 625–659, 2016.
- [16] S. Harnad, "The symbol grounding problem," *Physica D: Nonlinear Phenomena*, vol. 42, no. 1-3, pp. 335–346, 1990.
- [17] N. Schmitt, *Researching vocabulary: A vocabulary research manual*. Springer, 2010.
- [18] P. Prince, "Second language vocabulary learning: The role of context versus translations as a function of proficiency," *The modern language journal*, vol. 80, no. 4, pp. 478–493, 1996.
- [19] N. Schmitt, "Instructed second language vocabulary learning," *Language teaching research*, vol. 12, no. 3, pp. 329–363, 2008.
- [20] P. Joyce, "L2 vocabulary learning and testing: The use of L1 translation versus L2 definition," *The Language Learning Journal*, pp. 1–12, 2015.
- [21] Merriam-Webster, *Merriam-Webster's Collegiate Dictionary*, 11th ed., 2003.
- [22] J.-M. Poulin, "Stratégies efficaces pour l'apprentissage des mots d'un dictionnaire : Une approche basée sur les graphes," Master's thesis, Université du Québec à Montréal, 2018.
- [23] (2018). [Online]. Available: http://www.pearsonlongman.com/longman_france/pdf/dictionnaires.pdf
- [24] H. Jackson, *Lexicography: an introduction*. Routledge, 2013.
- [25] Merriam-Webster. (2018) Merriamwebsterthesaurus. [Online]. Available: <https://www.merriam-webster.com/thesaurus/>
- [26] D. A. Cruse, "The lexicon. the handbook of linguistics," 2002, ed. by Mark Aronoff and Janie Rees Miller, ch10, Oxford: Blackwell.
- [27] A. Polguère, *Lexicologie et sémantique lexicale : notions fondamentales*, 3rd ed., ser. Paramètres. Les Presses de l'Université de Montréal, 2016.
- [28] N. Gader, S. Ollinger, and A. Polguère, "One lexicon, two structures: So what gives?" in *Seventh Global Wordnet Conference (GWC2014)*. Global WordNet Association, 2014, pp. 163–171.
- [29] Oxford. (2018) Oxford English dictionary. [Online]. Available: https://en.oxforddictionaries.com/definition/us/word_form
- [30] TLFi. (2018) Trésor de la langue française informatisé. [Online]. Available: <http://www.cnrtl.fr/definition/dictionnaire>
- [31] A. Spencer, "The handbook of linguistics," in *The handbook of linguistics*, 1st ed., M. Aronoff and J. Rees-Miller, Eds. John Wiley & Sons, 2002, ch. Morphology, pp. 213–37.
- [32] C. Wenski-Béthoux, "Utilisation de produits multimédia pour la construction de compétences lexicale: analyse linguistique, psycholinguistique et didactique des apports des cédéroms, des sites internet et du travail en tandem pour l'apprentissage de l'allemand langue seconde," Ph.D. dissertation, Lyon 2, 2005.
- [33] F. De Saussure, *Cours de linguistique générale: Édition critique*. Otto Harrassowitz Verlag, 1916 (1989), vol. 1.
- [34] O. Duchacek, "L'homonymie et la polysémie," *Vox romanica*, vol. 21, p. 49, 1962.
- [35] D. Jurafsky and J. H. Martin, *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*, 2nd ed., ser. Prentice Hall series in artificial intelligence. Pearson Prentice Hall, 2009.
- [36] G. A. Miller, "Dictionaries in the mind," *Language and cognitive processes*, vol. 1, no. 3, pp. 171–185, 1986.
- [37] E. A. Corrêa, A. A. Lopes, and D. R. Amancio, "Word sense disambiguation: a complex network approach," *Information Sciences*, 2018.
- [38] R. Navigli, "Word sense disambiguation: A survey," *ACM Computing Surveys (CSUR)*, vol. 41, no. 2, p. 10, 2009.
- [39] R. V. Yampolskiy, "Turing test as a defining feature of ai-completeness," in *Artificial intelligence, evolutionary computing and metaheuristics*. Springer, 2013, pp. 3–17.
- [40] P. van Sterkenburg, *A practical guide to lexicography*. John Benjamins Publishing, 2003, vol. 6.
- [41] Wordsmyth. (2017) Wordsmyth. [Online]. Available: <https://www.wordsmyth.net>
- [42] D. McCarthy, R. Koeling, J. Weeds, and J. Carroll, "Finding predominant word senses in untagged text," *ACM*, 2004.
- [43] J. F. Sowa, *Knowledge representation logical, philosophical, and computational foundations*. Pacific Grove, Calif. ; Toronto: Brooks/Cole, 2000.
- [44] J. Hendler and F. van Harmelen, "The semantic web: webizing knowledge representation," *Foundations of Artificial Intelligence*, vol. 3, pp. 821–839, 2008.
- [45] S. Russell, P. Norvig, and A. Intelligence, "Artificial intelligence, a modern approach," *Artificial Intelligence. Prentice-Hall, Englewood Cliffs*, vol. 25, p. 27, 2010.
- [46] F. Lehmann, "Semantic networks," *Computers & Mathematics with Applications*, vol. 23, no. 2-5, pp. 1–50, 1992.
- [47] D. L. Nelson, C. L. McEvoy, and T. A. Schreiber, "The university of south florida word association, rhyme, and word fragment norms," 1999. [Online]. Available: <http://w3.usf.edu/FreeAssociation/>
- [48] J. A. Bondy, U. S. R. Murty *et al.*, *Graph theory with applications*. Oxford, UK: Elsevier Science Ltd., 1976.
- [49] V. V. Vazirani, *Algorithms d'approximation*. Traduction de: Algorithms, ser. Collection IRIS. Paris: Springer, 2006.
- [50] R. M. Karp, *Reducibility among Combinatorial Problems*. Boston, MA: Springer US, Miller, Raymond E. and Thatcher, James W. and Bohlinger, Jean D. editors, 1972, pp. 85–103.
- [51] H.-M. Lin and J.-Y. Jou, "On computing the minimum feedback vertex set of a directed graph by contraction operations," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 19, no. 3, pp. 295–307, 2000.
- [52] M. Lapointe, A. B. Massé, P. Galinier, M. Lord, and O. Marcotte, *Enumerating minimum feedback vertex sets in directed graphs*. LaBRI, Université Bordeaux 1, 2012, ch. Enumerating minimum feedback vertex sets in directed graphs, pp. 101–102.
- [53] C. D. Manning, P. Raghavan, and H. Schütze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.
- [54] J. Leskovec, A. Rajaraman, and J. D. Ullman, *Mining of massive datasets*. Cambridge university press, 2014. [Online]. Available: <http://i.stanford.edu/~ullman/mmds.html>
- [55] S. Harnad, "Symbol-grounding problem," *Encyclopedia of cognitive science*, 2003.
- [56] —, *To Cognize is to Categorize: Cognition is Categorization*. Elsevier, 2005.
- [57] A. L. Z. Monge, "L'évolution de l'enseignement du vocabulaire dans la classe de L2," *Revista de Linguas Modernas*, pp. 437–447, 2013.
- [58] I. S. Nation, *Making and using word lists for language learning and testing*. John Benjamins Publishing Company, 2016.

- [59] C. K. Ogden, "Basic English: A general introduction with rules and grammar, paul treber & co," *Ltd. London*, vol. 1940, 1930.
- [60] —. (2018) Ogden's basic english. [Online]. Available: <http://ogden.basic-english.org/wordmenu.html>
- [61] M. West, *A general service list of English words: with semantic frequencies and a supplementary word-list for the writing of popular science and technology*. Addison-Wesley Longman Limited, 1953. [Online]. Available: <http://jbauman.com/aboutgsl.html>
- [62] A. Coxhead, "A new academic word list," *TESOL quarterly*, vol. 34, no. 2, pp. 213–238, 2000.
- [63] V. Brezina and D. Gablasova, "Is there a core general vocabulary? introducing the new general service list," *Applied Linguistics*, vol. 36, no. 1, pp. 1–22, 2013.
- [64] C. Browne, "A new general service list: The better mousetrap we've been looking for," *Vocabulary Learning and Instruction*, vol. 3, no. 2, pp. 1–10, 2014.
- [65] C. Browne, B. Culligan, and J. Phillips. (2018) New general service list (ngsl). [Online]. Available: <http://www.newgeneralservicelist.org>
- [66] M. Brysbaert and B. New, "Moving beyond kučera and francis: A critical evaluation of current word frequency norms and the introduction of a new and improved word frequency measure for american English," *Behavior research methods*, vol. 41, no. 4, pp. 977–990, 2009.
- [67] K. Black, "Cambridge international dictionary of English," *The Booklist*, vol. 93, no. 19–20, June 1997.
- [68] Longman. (2018) Ldoce. [Online]. Available: <https://pearsonerpi.com/fr/elt/dictionaries/longman-dictionary-of-contemporary-english>
- [69] E. K. Brown and A. Anderson, Eds., *Encyclopedia of language and linguistics [ressource électronique]*, 2nd ed. Amsterdam: Elsevier, 2006. [Online]. Available: <https://www.sciencedirect-com.proxy.bibliotheques.uqam.ca:2443/science/referenceworks/9780080448541>
- [70] K. Toutanova, D. Klein, C. D. Manning, and Y. Singer, "Feature-rich part-of-speech tagging with a cyclic dependency network," in *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, ser. NAACL '03. Stroudsburg, PA, USA: Association for Computational Linguistics, 2003, pp. 173–180.
- [71] V. Batagelj, A. Mrvar, and M. Zaveršnik, *Network analysis of dictionaries*. University of Ljubljana, Inst. of Mathematics, Physics and Mechanics, Department of Theoretical Computer Science, 2002.
- [72] E. W. Weisstein. (2003) Graph diameter. [Online]. Available: <http://mathworld.wolfram.com/GraphDiameter.html>
- [73] S. E. Schaeffer, "Graph clustering," *Computer science review*, vol. 1, no. 1, pp. 27–64, 2007.
- [74] A. Hagberg, P. Swart, and D. S. Chult, "Exploring network structure, dynamics, and function using networkx," Los Alamos National Lab.(LANL), Los Alamos, NM (United States), Tech. Rep., 2008.
- [75] P. Bonin, A. Méot, L. Aubert, N. Malardier, P. Niedenthal, and M.-C. Capelle-Toczek, "Normes de concrétude, de valeur d'imagerie, de fréquence subjective et de valence émotionnelle pour 866 mots," *L'année Psychologique*, vol. 103, no. 4, pp. 655–694, 2003.
- [76] K. J. Gilhooly and R. H. Logie, "Age-of-acquisition, imagery, concreteness, familiarity, and ambiguity measures for 1,944 words," *Behavior research methods & instrumentation*, vol. 12, no. 4, pp. 395–427, 1980.
- [77] M. Coltheart, "The mrc psycholinguistic database," *The Quarterly Journal of Experimental Psychology Section A*, vol. 33, no. 4, pp. 497–505, 1981.
- [78] M. Wilson, "Mrc psycholinguistic database: Machine-usable dictionary, version 2.00," *Behavior research methods, instruments, & computers*, vol. 20, no. 1, pp. 6–10, 1988.
- [79] T. A. Harley, *The psychology of language: From data to theory*. Psychology press, 2013.
- [80] A. Paivio, J. C. Yuille, and S. A. Madigan, "Concreteness, imagery, and meaningfulness values for 925 nouns," *Journal of experimental psychology*, vol. 76, no. 1p2, p. 1, 1968.
- [81] C. Browne, B. Culligan, and J. Phillips. (2018) New academic word list (nawl). [Online]. Available: <http://www.newacademicwordlist.org>
- [82] C. Browne and B. Culligan. (2018) Business service list (bsl). [Online]. Available: <http://www.newgeneralservicelist.org/bsl-business-service-list/>
- [83] —. (2018) Toeic service list (tsl). [Online]. Available: <http://www.newgeneralservicelist.org/toeic-list/>
- [84] V. Kuperman, H. Stadthagen-Gonzalez, and M. Brysbaert, "Age-of-acquisition ratings for 30,000 English words," *Behavior Research Methods*, vol. 44, no. 4, pp. 978–990, 2012.
- [85] B. MacWhinney, *The CHILDES Project: Tools for Analyzing Talk*. Mahwah, NJ: Lawrence Erlbaum Associates, third edition, 2000.
- [86] M. Brysbaert, A. B. Warriner, and V. Kuperman, "Concreteness ratings for 40 thousand generally known English word lemmas," *Behavior research methods*, vol. 46, no. 3, pp. 904–911, 2014.
- [87] R. A. Amsler, "The structure of the merriam-webster pocket dictionary," Ph.D. dissertation, The University of Texas at Austin, 1980.
- [88] D. Bullock, "Nsm+ ldoce: A non-circular dictionary of english," *International Journal of Lexicography*, vol. 24, no. 2, pp. 226–240, 2010.

Subjective Assessment of Text Quality on Smartphone Display with Super Resolution

Aya Kubota

Department of Information Science
Kogakuin University
Tokyo, Japan
em17008@ns.kogakuin.ac.jp

Seiichi Gohshi

Department of Information Science
Kogakuin University
Tokyo, Japan
gohshi@cc.kogakuin.ac.jp

Abstract— It is only a decade ago that smartphones appeared on the market. However, the market has since grown rapidly, and people of all ages now use smartphones. In many cases, people read text on their smartphones, but depending on the design of a website, it may be difficult to read its text. By improving the resolution of the text, the readability of the text can be improved. One research area for increasing the resolution is Super Resolution (SR), which includes Non-Linear Signal Processing Super-Resolution SR (NLSP), a method that can be implemented on smartphones. However, NLSP has never been applied to text to improve readability. Text has many kinds of characters, such as Chinese characters, and alphabets of different languages. Features of these characters are different. We applied NLSP to Japanese text including Chinese characters, katakana and numbers, displayed on Liquid Crystal Display (LCD), and verified its effectiveness using a subjective assessment. In addition, we applied NLSP to English text and compared the difference between image quality text with and without NLSP. The subjective assessment results show that NLSP can increase the resolution of Japanese and English text. Thus, the assessment results for text on LCD are discussed in this paper.

Keywords- *Nonlinear signal processing; Super-Resolution; Subjective assessment; Smartphone.*

I. INTRODUCTION

Smartphones have become daily necessities in modern society. In addition to processing communication functions, such as telephone and e-mail, it is possible to obtain information in real time via the Internet. When used for the above functions, text must often be read, in the form of operation buttons or explanatory text. Support functions to make text easier to read, such as changing the font size, are set in the application that is preinstalled in the operating system (such as mail, smartphone settings, etc.). However, there are websites that do not have a font size larger than a certain size even if the text is enlarged, and sites where the color of the background and the text is very similar. Therefore, problems, such as these can make it difficult to read the text.

Improving the resolution of the images can make it easier to read a text. One method to improve resolution is Super-Resolution (SR) technology [1]. Most 4K Televisions (TV) are equipped with SR. Non-Linear Signal Processing SR (NLSP) is an SR technology that can be embedded into smartphones [2]. The algorithm is simple and fast: hence,

processing with software is possible, and smartphones with NLSP are already being sold on the market [3]. The effectiveness of NLSP is higher than that of other SR technologies [4][5], and NLSP is effective even in smartphone videos [6].

However, the effectiveness of NLSP for text on smartphone display has not been verified. In this study, we verify the effectiveness of using a smartphone with NLSP compared to one without NLSP.

Images processed with NLSP are introduced only to the display of the smartphone and there is no electric output of the processed image. Therefore, it is impossible to use an objective assessment because the objective assessment requires electric image signal with and without NLSP. Subjective assessment is the only way to assess the difference between the displays. However, subjective assessment is only a reflection of how we feel. It is difficult to ensure the reproducibility of subjective assessments. In addition the subjective assessments require observers and time to assess the image quality.

Although there are issues with the subjective assessment, the ITU-R has standardized subjective assessment methods. ITU-R BT.710 recommends experimental conditions to obtain reproducible results in subjective assessment experiments [7]. However, BT.710 does not mention practical quantitative scoring assessment, which is defined in BT.500. They are the Double Stimulus Continuous Quality Scale (DSCQS) and the Double Stimulus Impairment Scale (DSIS). In this experiment, we need to compare five smartphones and they are different manufactures products, and BT.500 and BT.710 do not meet our requirements. One of our authors developed a subjective assessment for multiple displays [7][8]. It applies best–worst method, and statistical analysis is introduced to analyze reproducibility. It shows good results if the images/videos are selected appropriately.

This paper is organized as follows. In Section II, the subjective assessment for multiple displays is explained. In Section III, NLSP is explained. In Section IV, the test images are presented and the experiments are explained. In Section V, the statistical analysis is adapted to the assessment results and in Section VI the analyzed result is discussed. Section VII is the conclusion of the paper.

II. QUANTITATIVE ASSESSMENT OF DIFFERENT SMARTPHONE DISPLAYS

Smartphone displays show us images as optical signal. It is difficult to compare image quality between different types of smartphones. It is very difficult to conduct objective quantitative assessments between optical images. Until now, objective comparison of the image quality using different types of smartphone displays has not been reported. In this study, we introduced subjective assessment, and made it possible to quantitatively assess the image quality between different displays.

Objective assessment and subjective assessment are evaluation methods. Objective assessments analyze the signal and express high and low image quality by a numerical value. However, the results of an objective assessment do not always match with how we feel. For example, an original image is given in Figure 1(a), and the degraded image is given in Figure 1(b). The Peak Signal to Noise Ratio (PSNR) of the degraded image in Figure 1(b) is 40.1112dB. A PSNR 40dB is generally said to be a high image quality [9]; however, Figure 1(b) contains degradation in the form of a black square in the center of the image. When images include local degradation, the results of PSNR sometimes deviate from our feeling.

Thus, objective assessments cannot reflect image quality accurately. In addition, objective assessments require a comparison of the assessment image with the original image. As discussed in the previous section, signals processed inside the smartphone cannot be output anywhere outside the display. Therefore, assessment by signal analysis is impossible, and thus the experiment is conducted using subjective assessment.

The best–worst method was adopted as the assessment method using multiple displays. Normalized ranking method and paired comparison method are other assessment methods. Experimental stimuli are ranked at once in the normalized ranking method. The process of the method is simple; however, when differences between the stimuli are small, sometimes the differences cannot be detected because of large differences between stimuli influences. In the paired comparison method, stimuli are compared one on one and ranked. Two stimuli are selected, and the observers evaluate the stimuli based on the other. Thus, differences between stimuli can be obtained in detail. However, evaluation is performed for all the stimulus combinations, which places a heavy burden on the observers. In the best–worst method, observers select the best stimuli and the worst stimuli. After excluding the selected stimuli, the observers again select the best and the worst from the remaining stimuli.

Although normalized ranking method is common, the best–worst method can detect small differences more accurately than the normalized ranking method. Accuracy of the best–worst method is lower than the paired comparison. However, the time consumption of the best–worst method is shorter than that of the paired comparison. It means that observers' burden of the best–worst method is lower than that of the paired comparison. Therefore, in this paper, the best–worst method is adopted.



Figure 1. Objective assessment by PSNR

In this study, an assessment experiment was conducted using five smartphones. The test images are screenshots of a website containing text.

III. SUPER RESOLUTION

Super resolution technology is a method to improve image/video resolution, and mounted on most 4K TVs. Although smartphones that has 4K resolution display are for sale, images/videos that have 4K resolution are insufficient. Therefore, it is necessary to improve the image/video resolution. However, it is impossible to mount current mainstream SR technologies to smartphones due to the technical reason. In this section, problems of the conventional SR technologies if they are mounted to smartphones, and NLSP, which can solve the issue are explained.

A. Super resolution for smartphones

The purposes of TV and smartphone are different; therefore, performance difference, such as display size and processing speed, is great.

If conventional SR are mounted to smartphones, issues will occur. For example, image quality difference cannot be understood on small smartphone displays, and processing will be slow because processing works on software. Although designed hardware for implementation SR is mounted in TVs, smartphones have no space to mount new hardware.

Therefore, it is impossible to implement SR for TVs to smartphones. The size of the monitor becomes an important factor in seeing an SR processed image [10]. Much research on SR has been conducted. However, it does not discuss the difference in clarity of the image depending on the display size. Even if images are processed with SR, whether SR is effective or not on small smartphone displays has not been reported. SR studies freely select their processed image sizes to recognize the resolution improvement. Personal Computer (PC) monitors have been used to check image resolution. Although commercial HDTV sets with SR technology can be used (Tos, 2009 [11]), the screen sizes of HDTVs are 40 inches or larger. On the other hand, the display sizes of commercial smartphones are approximately 5 to 6 inches. It is difficult to recognize improvement with SR on a small display. Even if we can recognize resolution improvement on a large display, such as a PC monitor or HDTV, it is not

always recognizable on smartphone displays. Therefore, if we are to implement SR technology, it is meaningless to implement the SR function unless resolution improvement is recognized. Smartphones are developed on the assumption that they are portable; therefore, the small devices are used to carry out many functions. Thus, it is impossible to add devices to a smartphone to use SR. There are two difficulties in implementing SR on a smartphone with limited resources. The first is the complexity of the SR algorithm. Many SR algorithms have been proposed (Farsiu et al. [10], Park et al. [12], Katsaggelos et al. [13], van Eekeren et al. [14], Panda et al. [15], Glasner et al. [16], Sun et al. [17], Dong et al. [18]). Super Resolution image Reconstruction (SRR) and Learning-Based Super Resolution (LBSR) are typical SR technologies, though many others have been proposed. However, all SR algorithms, including SRR and LBSR are difficult to use in real time for video because they require iteration to create a high-resolution image. Iteration is very time consuming and difficult to execute on the CPU/GPU of a smartphone. Although a non-iteration SRR algorithm for HDTV has been proposed (Matsumoto and Ida, 2010), the resolution is lower than that of a conventional HDTV and an additional device for implementation SRR is required.

The second difficulty is SR on smartphones must work on the CPU/GPU of a smartphone. Due to the space and power consumption, it is difficult to add a device for SR implementation to a smartphone. If we add a new device to a smartphone, the new parts will shorten battery duration owing to higher power consumption. Thus, to use SR on a smart-phone, it is necessary to work with the limited resources, such as the CPU/GPU, of a smartphone. The CPU/GPU executes many tasks, and resources, such as the memory bandwidth are limited. If sufficient CPU/GPU power and resources are not provided for the SR process, a video cannot be processed in real time, and frame drops can occur. In the worst case, the video will freeze. To overcome these difficulties, an SR algorithm for a smartphone must be simple and sufficiently light to work on CPU/GPU power and limited resources.

B. NLSP

NLSP is a simple and fast SR technique, which made it possible to implement SR to smartphones for the first time in the world.

The process is similar to enhancer that it increases resolution by emphasizing edges; however, NLSP emphasizes high-frequency components extracted from the input image using a nonlinear function [2]. Figure 2 shows the signal flow of NLSP. The input signal has two paths. The first path consists of a High-Pass Filter (HPF), Non-Linear Function (NLF), and a Limiter (LMT). This path generates high-frequency components that the original video does not have. High-frequency components include the edges and details of an image/video. HPF detects the edges of the input signal. Then, the detected edges are processed with the NLF. It can create high-frequency components not included the

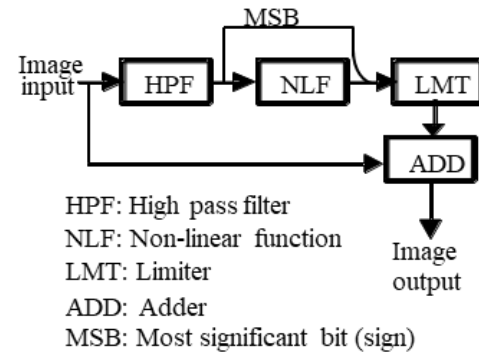


Figure 2. Block diagram of NLSP

input video. An example of an NLF is a cubic function ($f(x) = x^3$). The function can amplify the high-frequency components by as many as three times. We explain the NLF using the cubic function $f(x) = x^3$. It is well known that images and videos can be expressed by sine and cosine waves with Fourier series. If $f(x)$ is assigned $\sin\theta$, it is changed to $(\sin\theta)^3$ using the cubic function. Similarly, if $f(x)$ is assigned $\cos\theta$, it is changed to $(\cos\theta)^3$. $(\sin\theta)^3$ can be changed to $\sin(3\theta)$ and $(\cos\theta)^3$ can be changed to $\cos(3\theta)$. $\sin(3\theta)$ and $\cos(3\theta)$ are harmonic waves, and the frequency is higher than the original video. The cubic function is just an example of a nonlinear function, and the NLF is used to create the high-frequency components by harmonic waves. The harmonic waves are generated only from the edges detected with the HPF. Flat areas do not have edges; therefore, there are no harmonic waves. The LMT saturates these large values to fit the harmonic waves to the video.

The second path is from the input, and it is directly connected to the Adder (ADD). The ADD adds the harmonic waves processed by the LMT to the original video. The process is conducted pixel by pixel.

Therefore, the output of the ADD has high-frequency components not included the original video. This processing method can improve the resolution, and even generate high-frequency components that exceed the Nyquist frequency of the original video. This simple and fast algorithm has led to the development of real-time NLSP hardware.

Figure 3 shows an image processed with NLSP hardware. Figure 3(a) is an enlarged image from HDTV to 4K. Figure 3(b) is the NLSP processed result of Figure 3(a). Although Figure 3(a) is blurry, Figure 3(b) more clearly expresses the edge and details than Figure 3(a). Figures 3(c) and (d) are the two-Dimensional Fast Fourier Transform (2D-FFT) results of Figures 3(a) and (b) respectively.

Figures 3(c) and (d) show the frequency characteristics in the frequency domain. The horizontal and vertical axis are the horizontal and vertical frequencies of the image. The center of the image shows low-frequency. The frequency is higher with distance from the center. Figure 3(d) has horizontal and vertical high-frequency components that are

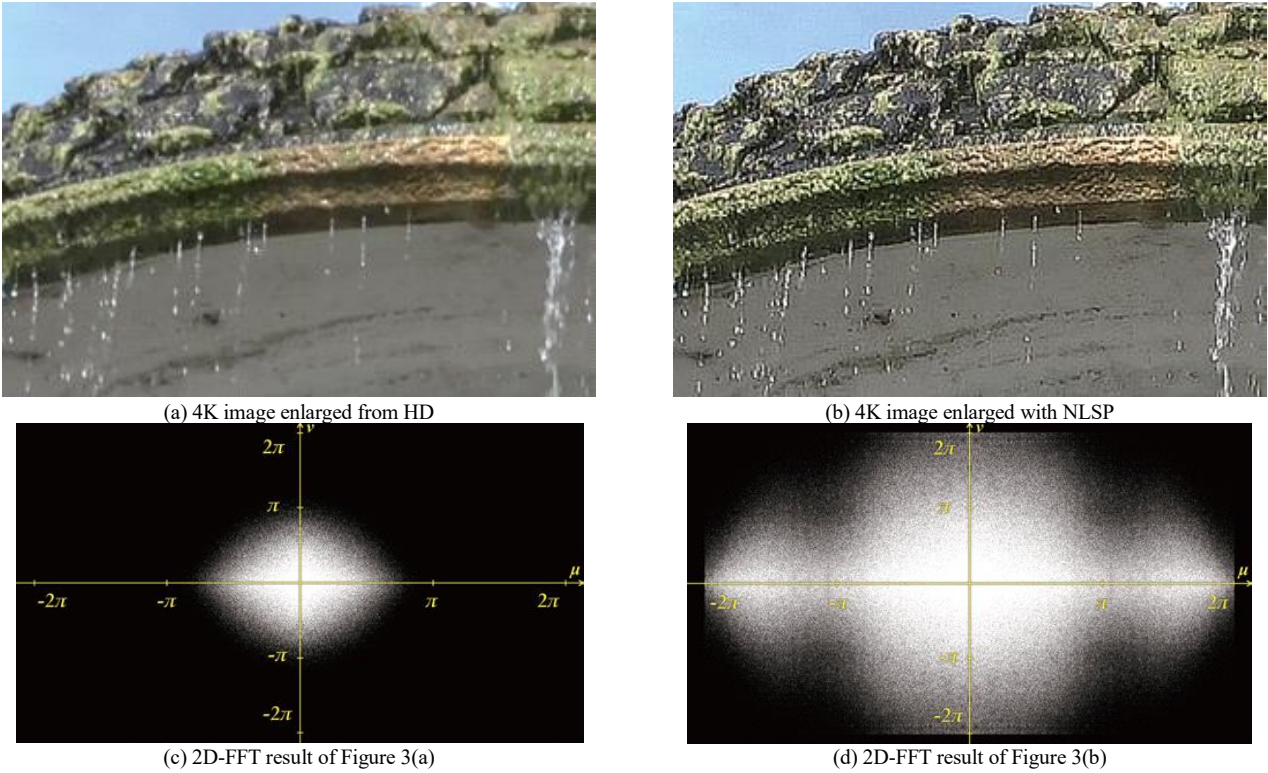


Figure 3. Image processed with real-time NLSP hardware

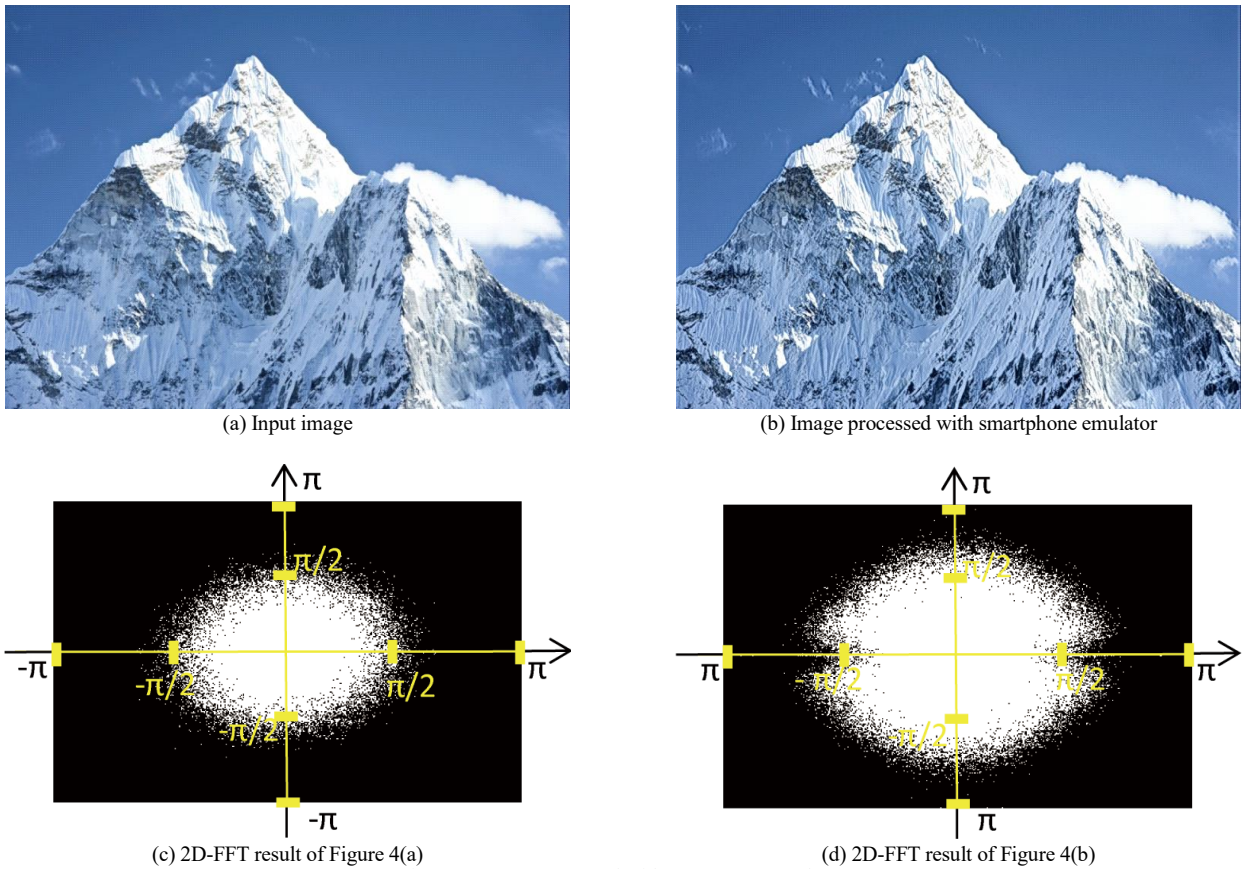


Figure 4. Image processed with an NLSP smartphone

not present in Figure 3(c). This means that NLSP creates high-frequency components, and increases the resolution.

Although Super-Resolution image Reconstruction (SRR) and Learning-Based Super Resolution (LBSR) are the current mainstream SR technologies, they cannot be mounted to a smartphone. SRR is a technology that generates a high-resolution image from multiple degraded images [10]; however, the processing requires iteration. When the input image and output image have the same resolution, the technique is not very effective [19]. LBSR is a method that increases resolution using a database [20]. The effectiveness is affected by the database, and the processing requires both an expensive database and iteration. Thus, both of the above technologies require complex processing. In addition, their effectiveness is lower than that of NLSP [5][8].

Although the NLSP algorithm is very simple, whether NLSP can process videos in real-time on CPU/GPU of a smartphone has not been verified. One of the authors used a smartphone emulator to prove that NLSP can work normally on a smartphone [2]. Figure 4 shows the NLSP processed result with a smartphone emulator. Figure 4(a) shows a frame of an input video. Figure 4(b) shows the NLSP processed result of Figure 4(a) on a smartphone. Figure 4(a) more clearly expresses the edge and the details than Figure 4(b) does. Figures 4(c) and (d) show the 2D-FFT results of Figures 4(a) and (b) respectively. Figure 4(d) has horizontal and vertical high-frequency components not included Figure 4(c). The results show that NLSP can process in real-time and improve the resolution on a smartphone. A smartphone with NLSP has already been sold on the markets (Figure 5) [3].

IV. EXPERIMENT

The effect of image processing differs, depending on the image. We adjusted NLSP for text; hence, it was necessary to verify the effect of NLSP for text. When a new technology is developed, it is necessary to compare a processed image with an unprocessed image. Thus, in this experiment, a smartphone with NLSP and one without NLSP were compared. The result of the comparison indicates the effects of using NLSP. In addition, the experiment was conducted using smartphones from different manufacturers and verified the effect of NLSP in comparison with other technologies.

Text includes many types of characters, such as Chinese characters, hiragana, and alphabets. Each character has different features. Most Chinese characters consist of straight lines. Hiragana and alphabets consist of straight and curved lines. Therefore, even if NLSP is effective when applied to Chinese characters, we do not know whether NLSP is effective or not for characters that have different features. Thus, in this study, we conducted a subjective assessment to evaluate the image quality of Japanese text including Chinese characters, katakana, and hiragana. After, a similarly subjective assessment was conducted using English text.

A. Experimental Condition

The observers were instructed on the experimental procedure, the meaning of resolution and the point of evaluation. Explanation of the resolution was conducted



Figure 5. Developed Smartphone with NLSP

using training images to make the observers understand correctly. In addition, the observers were instructed not to consider the color, brightness, or noise of the image. When the observers purchase a smartphone, the viewing distance is different for each observer. Thus, observers could freely adjust the viewing distance. After evaluation, we investigated points where the observers gazed to judge whether the observers correctly evaluated differences in resolution.

B. Test Images

Nine screenshots of websites containing text were used as experimental images. Five images were Japanese text, and the others were English text. Japanese text images included hiragana, katakana, and Chinese characters. The images are shown in Figure 5. The resolution of all the images is WQHD. Figure 5 [a]-[e] shows the Japanese text images. Figure 5 [f]-[i] shows the English text images. The Japanese text images are of websites browsed by many people (a site for smartphones, a PC, a map). The site for smartphones is enlarged and viewed when the site has small text; therefore, an un-enlarged site image and two enlarged site images were used. One of the two enlarged images contained text with only small differences in color from the background color. Similarly, the three English text images are screenshots of websites for a smartphone and a map. When web articles written on PDF are browsed, the resolution is often low. Thus, one of the website screenshots for smartphones is a PDF article page.

C. Observers

At least 20 observers are required for adequate statistical analysis. In this experiment, 23 observers participated in the experiment and had normal visual acuity and color vision. Non-experts who do not work in the image industry cannot always distinguish image quality differences, even if experts can distinguish them. If there is a significant difference in the experiment using non-experts, the difference of image quality is great. Therefore, all the observers were non-experts.



Figure 6. Test images

D. Experiment 1

In Experiment 1, NLSP was applied to Japanese text, and the subjective assessment to evaluate the image quality of NLSP for text was conducted. A smartphone with NLSP and a smartphone without NLSP, and different manufacturer's smartphones were used. This experiment shows that NLSP is more effective than the conventional SR technologies.

1) Experimental Equipment

Five smartphones were used in this experiment. To ensure that the results were not caused by display differences, two of the five smartphones featured the same terminal. One was a smartphone with NLSP (smartphone A), and the other was one without NLSP (smartphone B). The remaining three smartphones were smartphones from different manufacturers (smartphone C–E). The display resolution of smartphones A and B was WQHD (2560 × 1440), whereas that of the others was full HD (1920 × 1080). The brightness was adjusted to be close to the same brightness.

2) Experimental Method

The observers evaluated the image quality of the test image and ranked the five smartphones by resolution. The

best-worst method was used in the experiment. First, the observers selected the best (1st rank) and the worst (5th rank) smartphones from the five smartphones. Second, the next best (2nd rank) and the next worst (4th rank) smartphones were selected in the same way from the remaining three smartphones. The remaining smartphone was ranked 3rd.

E. Experiment 2

In Experiment 2, English text images with and without NLSP were compared and evaluated.

1) Experimental equipment

Two smartphones were used to evaluate the image quality. These were the same terminal smartphones. One smartphone output images processed with NLSP to a display. The other output unprocessed images. When the image quality assessments were conducted using multiple displays, it was proven that an individual difference of displays did not affect the results [21].

The smartphones have 5.4 inch display, and the resolution is WQHD. The brightness was adjusted to be close to the same brightness on both devices.

The observers compared images displayed on smartphones, and chose the smartphone, which had the higher

TABLE I. Analysis result (Map)

l/k	R_l	f_{kl}					P_l	ε_l	$K_{\varepsilon l}$
		A	B	C	D	E			
1	5	22	1	0	0	0	90	0.1	1.28
2	4	1	2	3	17	0	70	0.3	0.52
3	3	0	9	9	2	3	50	-0.5	0.00
4	2	0	4	8	4	7	30	-0.3	-0.52
5	1	0	7	3	0	13	10	-0.1	-1.28
$\sum (f_{kl} \times K_{\varepsilon l})$		28.72	-8.74	-6.47	6.82	-20.33			
R_k		1.25	-0.38	-0.28	0.30	-0.88			
S_k^2		0.15	0.71	0.52	0.40	0.48			

resolution. To prevent prejudice from affecting the results, the state of NLSP (ON/OFF) was not revealed to the observers.

V. RESULTS

In this section, the results of the two experiments in the previous section are explained.

1) Experiment 1 results

The assessment results were analyzed, and the presence or absence of significant differences was identified. The assessment results were quantified, and the average scores representing the image quality of each stimulus were calculated [22]. The calculation requires a normalized score $K_{\varepsilon l}$, which can be calculated using P_l and ε_l . P_l is the average of each segment of the range from 0 to 100 separated into the number of stimuli. In this experiment, the number of stimuli, i.e., the number of smartphones (n), equals 5. The value ε_l is the median of each segment of the standard normal distribution separated into n segments. $K_{\varepsilon l}$ is the percentile of the standard normal distribution. Thus, $K_{\varepsilon l}$ is the distance from the average of the standard normal distribution. The values of $K_{\varepsilon l}$ were given as a normalized score according to rank. The average scores of the total score are the evaluation values for each stimulus.

The aggregate results of “Map” (Figure 3(a)) are shown in Table 1. The rows represent rank, and the columns represent stimuli (smartphones A–E). The values of intersection (f_{kl}) are the number of observers for stimulus k for rank l . Thus, f_{1A} indicates that 22 observers ranked the smartphone with NLSP (smartphone A) 1st.

First, rank is converted to a value. The higher the ranking, the higher the r_l value of the smartphone, where r_l is calculated as follows:

$$r_l = n - l + 1 \quad (1)$$

The percentile values P_l are calculated using r_l as follows:

$$P_l = \frac{r_l - 0.5}{n} 100 \quad (2)$$

The calculation results are shown in each row r_l , P_l of Table 1. Next, ε_l is calculated using (3) or (4). If the value of P_l is larger than 50, formula (3) is used. If the value of P_l is 50 or less, formula (4) is used. This is because the values of ε_l are calculated based on the point of the variance 0 of the standard normal distribution.

$$\varepsilon_l = 1 - \frac{P_l}{100} \quad (P_l > 50) \quad (3)$$

$$\frac{P_l}{100} \quad (P_l \leq 50) \quad (4)$$

The calculation results are shown in row ε_l of Table I. $K_{\varepsilon l}$ is calculated using ε_l from the normal distribution table. The values of $K_{\varepsilon l}$ shown in Table I were given to each stimulus according to the ranking. The average scores (R_l) of the total scores ($\sum (f_{kl} \times K_{\varepsilon l})$) are the evaluation values of the stimulus. For example, the average score R_A is calculated as follows: $R_A = 28.72/23 \approx 1.25$. The average scores and total scores are shown in Table 1. The average scores of “Map” (Figure 3(a)) are shown in the yardstick graph in Figure 4. The horizontal axis indicates the average score. The marks on the axis (oval, triangle, square, rhombus, and x) indicate the average scores of each stimulus (smartphone A, smartphone B, smartphone C, smartphone D, and smartphone E, respectively). The higher the average score, the higher the evaluation. In Table 1, the average score of smartphone A is the highest, indicating that smartphone A has the highest resolution.

A t-test was used to verify the significant difference between the stimuli. The variance of the average score (S_k^2) and the statistical quantity t_0 are calculated as follows:

$$S_k^2 = \frac{\sum \{f_{kl} \times (K_{\varepsilon l})^2\}}{\sum (f_{kl})} - R_k^2 \quad (5)$$

$$t_0 = \frac{R_x - R_y}{\sqrt{\sum(f_{kl}) (S_x^2 + S_y^2)}} \sqrt{\sum(f_{kl}) \sum \{(f_{kl}) - 1\}} \quad (6)$$

The value $\sum(f_{kl})$ indicates the number of observers. x and y are stimuli. The calculation results are shown in Table 1. The values of t are calculated using the Degree of Freedom (DoF) from t distribution. In this experiment, the DoF is $\text{DoF} = 2 * \sum(f_{kl}) - 2 = 46 - 2 = 44$. The t value of 1% significant level is $t_{1\%} = 2.414134$ and that corresponding to a 5% significant level is $t_{5\%} = 1.68023$. If the value of t_0 is larger than the value of $t_{5\%}$, there is a significant difference between stimuli.

Here, smartphone A is the highest, and smartphone D is the second highest. The t_0 value between smartphones A and D ($t_0(A, D)$) and the result of the t -test is as follows:

$$t_0(A, D) = 10.33 > t_{1\%} \quad (7)$$

In (7), $t_0(A, D)$ is larger than $t_{1\%}$. This result indicates that smartphone A has a higher resolution than smartphone D and has a significance value of 1%. The results of the 3rd rank (smartphone C), 4th rank (smartphone B), and 5th rank (smartphone E) are as follows:

$$t_0(D, C) = 4.13 > t_{1\%} \quad (8)$$

$$t_0(C, B) = 0.53 > t_{1\%} \quad (9)$$

$$t_0(B, E) = 2.77 < t_{5\%} \quad (10)$$

$t_0(D, C)$ and $t_0(C, B)$ are larger than $t_{1\%}$. Therefore, there are significant differences of 1% between smartphones D and C, and smartphones C and B. $t_0(B, E)$ is less than $t_{1\%}$ and $t_{5\%}$, indicating that there is no significant difference between smartphones B and E. The arrows indicate significant differences in the graph in Figure 4. The asterisks represent the level of significant difference between stimuli. “***” represents a significant difference of 1%, and “*” represents a significant difference of 5%. The analysis results of images [b–e] are shown in Figure 3 (b–e). Smartphone A has the highest resolution and significant differences of 1% between other smartphones in all the images. On the other hand, smartphone E has the worst resolution for all of the images and significant differences for four out of five images with the other smartphones.

2) Experiment 2 results

In Experiment 2, the observers compared the images processed with and without NLSP, and chose the smartphone, which had the higher resolution. We calculated ratio of each smartphones selected, and evaluated the statistical significant differences between the stimuli. In statistics, there are two important criteria about the significant difference. They are 95% and 99%. To obtain the 95% significant difference, at least 20 observers are required. If one out of twenty observers selects the smartphone with NLSP, the 95% significant difference between the stimuli is obtained. In contrast, if two observers select the smartphone with NLSP, the probability is 90%. In statistics, 90% does not indicate a significant difference. In this experiment, more than 20 observers participated. Thus, if 95% of the observers assess that the smartphone with NLSP has a higher resolution than the smartphone without NLSP, there is a significant difference of 95%.

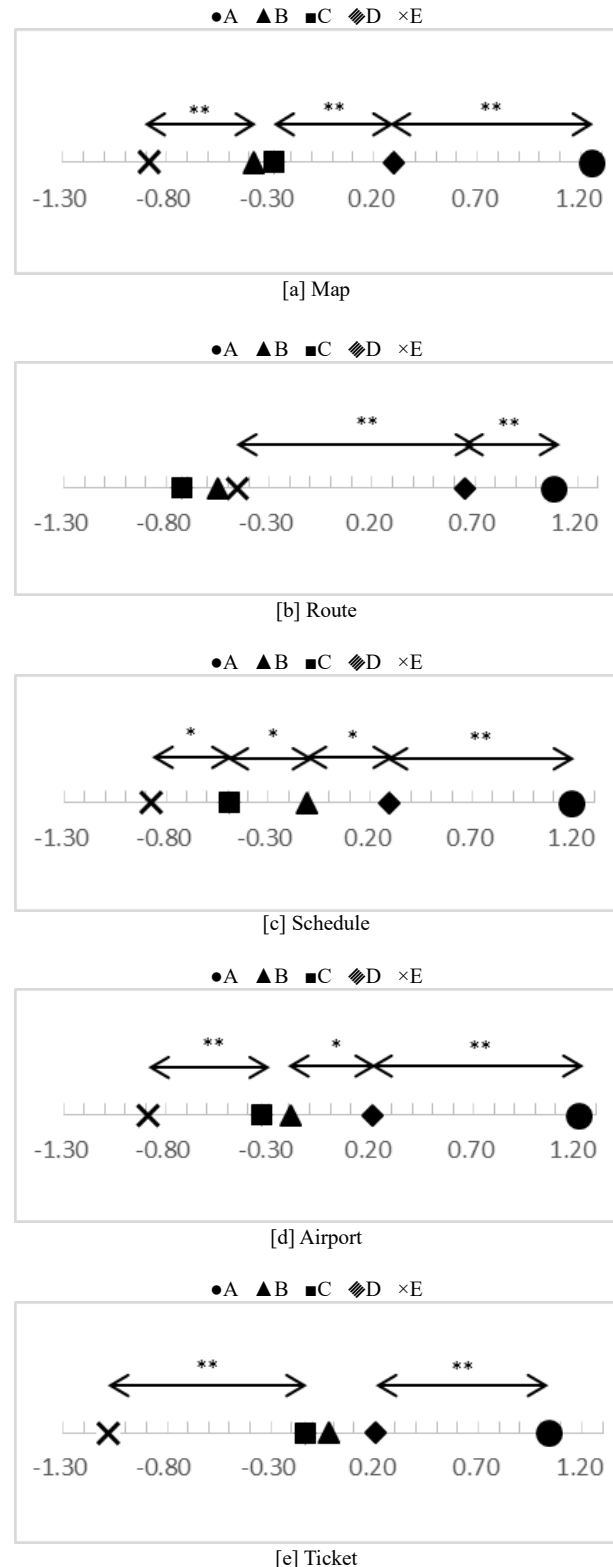


Figure 7. Assessment results (Experiment 1)

Figure 7 shows the results. The vertical axis represents the stimuli, the horizontal axis represents the number of observers. Here, the smartphone with NLSP is represented as

“NLSP,” and the smartphone without NLSP is represented as “OFF.”

The results are explained using the result of “Station” (Figure 7[a]) and that of “Company” (Figure 7[b]). The graph of Figure 7[a] indicates that 23 observers selected NLSP as having high resolution, and no one selected OFF. The results show that all the observers, that is, 100% observers assessed that NLSP has a higher resolution than OFF. Therefore, NLSP has a higher resolution, and there is a significant difference of 99% between the stimuli. In Figure 7[b], 22 observers selected NLSP, and one observer selected OFF. More than 95% of observers selected NLSP, which indicates that the result has reproducibility of more than 95%. Figure 7 [c], [d] show the results of the other test images. The results show that NLSP has a higher resolution than OFF, and there are significant differences of more than 95% between the stimuli for all of the images.

VI. DISCUSSION

In Experiment 1, smartphone A (with NLSP) has the highest score and a significant difference of 1% between the other smartphones (which are either without NLSP or from different manufacturers) in all the images. The results indicate that NLSP is valid for text on smartphone displays. The same results were obtained for all the images. Thus, NLSP is valid for images other than the five images used in this paper. There are significant differences between smartphones without NLSP. It is assumed that the results were influenced by the internal processing differences.

In Experiment 2, the smartphone with NLSP has a higher resolution than the smartphone without NLSP, and there are statistical significant differences of 1% or 5% between the stimuli for all scenes. Significant differences were obtained in both Japanese and English texts containing characters with different features. Therefore, it is assumed that NLSP can improve the resolution of text on smartphone displays.

In this experiment, a gazing point was not specified for the observers. In addition, there were significant differences in all of the images when all the observers were non-experts. From the above, there are clear differences of image quality between the images with NLSP and those without NLSP.

The same results were obtained in Experiments 1 and 2. Therefore, different of effect according to language cannot be found. Although we cannot technically specify the font type of test images, bold letters may affect the subjective assessment results. However, it can be adjusted by parameter controls.

VII. CONCLUSIONS AND FUTURE WORK

Subjective assessments using smartphones with NLSP and smartphones without NLSP were conducted to verify the effectiveness of NLSP for texts. In Experiment 1, Japanese text including hiragana, katakana, and Chinese characters was used as test images. In Experiment 2, test images included English text images.

The results of Experiments 1 using five smartphones indicated that the image quality of a smartphone with NLSP is the highest, and there are significant differences between the other smartphones. In Experiment 2, the images with and

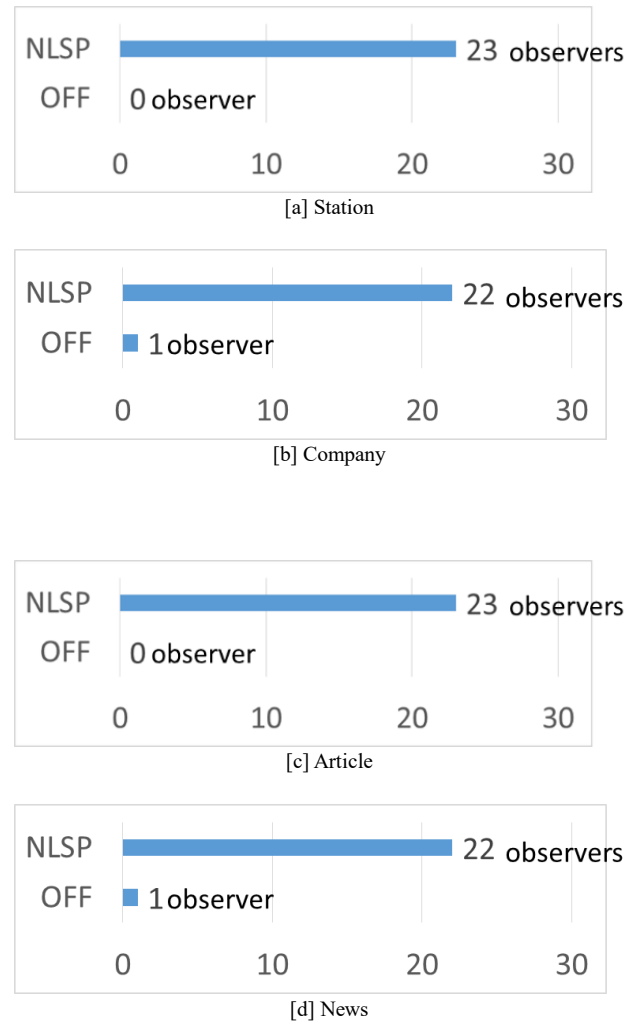


Figure 8. Assessment results (Experiment 2)

without NLSP were compared using two smartphones. The results show that the smartphone with NLSP has a higher resolution than the smartphone without NLSP does, and there were significant differences for all test images.

From the above, it was proven that NLSP can improve the resolution of texts on smartphone displays.

The statistical analyses indicate that the experimental results are reproducible. The conclusion that a smartphone with NLSP has the highest image quality was obtained for all the images; therefore, both the assessment method and the analysis method in this experiment were valid as subjective assessment methods.

In future work, we will apply to more characters and type of fonts, such as bold letters, and verify the general performance. Although the NLSP has been implemented only to one model of smartphone, its processing can work regardless of the operating system. Therefore, our final target is to implement the NLSP on as many smartphone models as we can.

REFERENCES

- [1] A. Kubota and S. Gohshi, "Subjective Assessment for Text with Super Resolution on Smartphone Displays," International Conference on Creative Content Technologies 2018 (CONTENT 2018), pp. 24-29, Feb. 2018.
- [2] S. Gohshi, "A new signal processing method for video: reproduce the frequency spectrum exceeding the Nyquist frequency," MMSys '12 Proceedings of the 3rd Multimedia Systems Conference, pp. 47-52, Sep. 2012.
- [3] http://www.fmworld.net/product/phone/f-02h/display.html?fmwfrom=f-02h_index. [retrieved: Nov. 2018].
- [4] H. Shoji and S. Gohshi, "Subjective Assessment for Resolution Improvement on 4K TVs: Analysis of Learning-Based Super-Resolution and Non-Linear Signal Processing Techniques," The Eleventh International Multi-Conference on Computing in the Global Information Technology (ICCGI2016), Vol. 2016, pp. 10-15, 2308-4529, Nov. 2016.
- [5] M. Sugie, S. Gohshi, H. Takeshita, and C. Mori, "Subjective Assessment of Super-Resolution 4K Video using Paired Comparison," 2014 International Symposium on Intelligent Signal Processing and Communication Systems (ISPACS-2014), pp. 42-47, Dec. 2014.
- [6] C. Mori and S. Gohshi, "Image Quality of a Smartphone Display with Super-Resolution," Proceedings of the ISCIE International Symposium on Stochastic System Theory and its Applications, Vol. 2016, pp. 340-344, 2016.
- [7] ITU-R. BT.710-4, "Subjective Assessment Method for Image Quality in High-Definition Television," Nov. 1998.
- [8] H. Shoji and S. Gohshi, "Subjective Assessment for Learning-Based Super-Resolution and Non-Linear Signal Processing Techniques," The 47th ISCIE International Symposium on Stochastic Systems Theory and Its Applications (SSS' 15), pp. 79-80, Dec. 2015.
- [9] Y. Mukaigawa, K. Suzuki, and Y. Yagi, "Analysis of subsurface scattering under generic illumination," 2008 19th International Conference on Pattern Recognition (ICPR), pp. 1-5, 1051-4651, Dec. 2008.
- [10] S. Farsiu and M. Dirk Robinson, "Fast and Robust Multi-Frame Super-Resolution," IEEE Trans Image Process 2004, Vol.13, no.10, pp. 1327-1344, 1941-0042, Oct. 2004.
- [11] Toshiba (in Japanese), <http://www.toshiba.co.jp/regza/detail/superresolution/resolution.html>. [retrieved: Nov. 2018].
- [12] S. H. Park, M. K. Park, and M. G. Kang, "Super-Resolution Image Reconstruction: A Technical Overview," IEEE Signal Processing Magazine, 1053-5888/03, pp. 21-36, May. 2003.
- [13] M. R. Banha, and A. K. Katsaggelos, "Digital Image Restoration," IEEE Signal Processing Magazine, 1558-0792, Vol. 14 (2), pp. 24-41, Mar. 1997.
- [14] A.W.M.Eekeren, K. Schutte, and L. J. van Vliet, "Multiframe Super-Resolution Reconstruction of Small Moving Objects," IEEE Transactions on Image Processing, 1941-0042, pp. 2901-2912, Vol. 19, no. 11, Nov. 2010.
- [15] S. Panda, M. S. R. S. Prasad, and G. Jena, "POCS Based Super-Resolution Image Reconstruction Using an Adaptive Regularization Parameter," International Journal of Computer Science Issues, Vol. 8, issue 5, No. 2, 1694-0814, Sep. 2011.
- [16] D. Glasner, S. Bagon, and M. Irani, "Super-Resolution from a Single Image," International Conference on Computer Vision (ICCV), pp. 349-35, 2380-7504, Oct. 2009.
- [17] J. Sun, Z. Xu, and H. Y. Shum, "Image Super-Resolution using Gradient Profile Prior," IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1-8, 1063-6919, Jun. 2008.
- [18] C. Dong, C. C. Loy, K. He, and X. Tanh, "Learning a Deep Convolutional Network for Image Super Resolution", European Conference on Computer Vision (ECCV), Vol. 8692, pp. 184-199, 978-3-319-10593-2, Sep. 2014.
- [19] S. Gohshi and I. Echizen, "Limitation of Super Resolution image Reconstruction," The Journal of The Institute of Image Information and Television Engineers, Vol. 36, No. 48, ME2012-125, pp. 1-6, 978-9-8975-8129-8, Nov. 2012.
- [20] W.T. Freeman, T.R. Jones, and E.C. Pasztorm, "Example-Based Super-Resolution," IEEE Computer Graphics and Applications," Vol. 22, No. 2, pp. 56-65, 1558-1756, Mar. 2002.
- [21] C. Mori, M. Sugie, H. Takeshita, and S. Gohshi, "Method of Subjective Image Quality Assessment of Multiple Displays," The Journal of the Institute of Image Electronics Engineers of Japan, Vol. 46, No. 2, pp. 315-323, Apr. 2017. (in Japanese)
- [22] T. Fukuda and R. Fukuda, "Ergonomics handbook," Tokyo: Scientist press co.ltd, pp. 41-71, 2009. (in Japanese).

Multi-dimensional Context Creation Based on the Methodology of Knowledge Mapping

Claus-Peter Rückemann

Westfälische Wilhelms-Universität Münster (WWU), Germany;
Knowledge in Motion, DIMF, Germany;
Leibniz Universität Hannover, Germany
Email: ruckema@uni-muenster.de

Abstract—The extended research presented in this paper focusses on advanced context creation as practicable in knowledge based disciplines. This extended research presents methods for multi-dimensional context creation based on the new methodology of Knowledge Mapping. The methodology of Knowledge Mapping allows the knowledge based mapping of objects and entities for purpose of context creation. The mapping to new context can improve complex knowledge mining, discovery, and decision making results. The new context increases the potential for creating new insights. The paper presents implemented methods and case studies along with an extended introduction of the new methodology used with advanced knowledge mining and provides the latest results of the present research. The different methodology based knowledge mining method implementations deploy various spatial representations for illustration. All method implementations utilise commonly available unstructured data and create new multi-disciplinary knowledge context. Resulting entities are spatially mapped. The results can be used for further analysis in integration with data and advanced tools, e.g., automated and visual analysis. The methodology can employ integrated knowledge resources and services for mapping support and can be applied to any content from arbitrary disciplines. The results of the mapping to new context can be used for knowledge mining workflows, for gaining new insight, and for creating and further improving long-term knowledge resources. The methodology also supports automated learning processes. This extended research aims on illustrating the flexibility of possible methods for new practical mining procedures and algorithms from the knowledge perspective.

Keywords—*Methodology of Knowledge Mapping; Data-centric Knowledge Mining; Multi-dimensional Context Creation; Spatial Knowledge Mapping Methods; High End Computing based on Knowledge Resources.*

I. INTRODUCTION

Knowledge Mining is a goal, which is required for a large number of application scenarios but which is nevertheless in practice widely based on plain methods of data mining.

This extended research is based on the new Methodology of Knowledge Mapping, which was presented at the GEOProcessing 2018 conference in Rome, Italy [1]. The paper goes beyond plain methods and the limited view of ‘data’.

The research presented here illuminates the superordinate knowledge view [2] and is therefore not restricted to a simple data view and focusses on advanced context creation for arbitrary knowledge. This paper presents context-methods for multi-dimensional context creation based on the new

methodology of Knowledge Mapping. The methodology of Knowledge Mapping allows the knowledge based mapping of arbitrary objects and entities for purpose of context creation. The mapping to new context can improve complex knowledge mining, discovery, and decision making results. The new, multi-dimensional context increases the potential for creating new insights. In terms of knowledge, context creation is a multi-disciplinary effort however limited and strict the discipline focus is defined.

The implemented context-methods and case studies along with of the illustrated case studies based on the new methodology are used with based on advanced knowledge mining and provide the latest results of the present research. The different methodology based knowledge mining method implementations deploy various spatial representations.

All the context-method implementations utilise commonly available unstructured data and create new multi-disciplinary knowledge context. Resulting entities are spatially mapped. The results can be used for further analysis in integration with data and advanced tools, e.g., automated and visual analysis.

The rest of this paper is organised as follows. Sections II and III introduce to the state of the art from previous work, fundamentals and motivation. Section IV introduces the new methodology of knowledge mapping. Section V discusses fundamentals, components, and used resources. Section VI presents the principles of multi-dimensional context creation based on knowledge mapping. Section VII illustrates implemented methods, generated for interactive dynamical context examples. Sections VIII and IX discuss the multi-dimensional features of methodology and implementation and summarise the lessons learned, conclusions, and future work.

II. STATE OF THE ART OF PREVIOUS WORK

It is a truth universally acknowledged, that any knowledge, e.g., based on unstructured and structured data, can contain parts, which may refer to other knowledge but which are not explicitly linked. Further, existing methods promising to deal with lexical and term mapping or ontologies showed deficient and inadequate for coping with challenges of arbitrary knowledge mapping and multi-dimensional context. Methods [3] and implementations for automated mapping [4] are not sufficient, the more as approaches do not span disciplines [5]. Term identification [6] is not suitable for mapping beyond simple context like bibliographic data, too. Available mapping

approaches are very limited to non-general knowledge related tasks [7], even when dealing with context [8].

Regarding managerial aspects, modern knowledge organisation systems [9] can support the processes [10], nevertheless, they are just components — besides knowledge. If system components are not specialised on knowledge itself but more or less on functional processes, e.g., tool components for collaboration [11], collaborative knowledge [12], and ‘knowledge based collaboration’ [13]. An approach with historical data from many multi-disciplinary sources is the Venice Time Machine (VTM) [14] project. However, there is no general methodological approach associated with the project.

The essential fundament for knowledge mapping is the knowledge. The methodology employed here was developed in order to create methods for the identification of entities inside of or referenced with data and create new context for knowledge objects and entities. Besides the context relevant for this research, further basic terms and definitions are explained in the referenced publications, e.g., for data entities, mapping, and computing [15] as well as for entities and references [16].

The principle of this approach is superior to data and information based approaches as the methodology takes benefit of knowledge complements [17]. Knowledge is an excellent integrator as it can complement, e.g., from [18]

- factual,
- conceptual,
- procedural,
- and metacognitive knowledge.

Data and information can be associated with all the complements. In consequence, this methodology allows to deploy a knowledge based level, e.g., creating knowledge mining where information can result from information peeling processes [17].

Knowledge mapping is the process of creating mappings between two data objects. In that way knowledge mapping contributes significantly to data integration and data sciences methods [19]. The means of referring objects and sub-objects, “entities”, with a new context is considered as “knowledge mapping”. Objects, e.g., a document, a part of a text, or an image may be associated with other objects, by its knowledge, e.g., its factual or conceptual knowledge. For example, creating new spatial context for textual entities in knowledge objects requires to build non-fixed associations, apply a fuzzy spatial locate, and implement a text location to map-mapping. The procedure enables to automatically create a spatial mapping for possible locations in a document, e.g., Points Of Interest (POI) or other places in a data set or file.

III. FUNDAMENTS AND MOTIVATION

The fundaments of terminology and understanding knowledge are layed out by Aristotle [20], being an essential part of ‘Ethics’ [21], which makes Aristotle probably most the primarily relevant knowledge reference. Information science can very much benefit from Aristotle’s fundaments and a knowledge-centric approach [18] but information science needs to go beyond the available technology-based approaches for building holistic and sustainable solutions, supporting a modern definition of knowledge [22]. Triggered by the results of a

systems cases study, it is obvious that superordinate systematic principles are still widely missing. Making a distinction and creating interfaces between methods and the implementation applications [23], the results of this research are illustrated here along with the practical example of the Knowledge Mapping methodology [1] enabling the creation of new object and entity context environments, e.g., implementing methods for knowledge mining context. This motivating background allows to build methods for knowledge mapping on a general methodological fundament.

The Organisation for Economic Co-operation and Development (OECD) has published principles and guidelines for access to research data from public funding [24]. The principles and guidelines are meant to apply to research data that are gathered using public funds for the purposes of producing publicly accessible knowledge. Anyhow, from the knowledge management point of view they have much wider importance as they

- address the protection of intellectual property,
- deal with knowledge generated from the re-use of existing data, and
- describe important aspects when establishing evaluation criteria.

The guidelines recommend the following items should be considered in establishing evaluation criteria:

- Overall public investments in the production and management of research data.
- Management performance of data collection and archival agencies.
- Extent of re-use of existing data sets.
- Knowledge generated from the re-use of existing data.
- The use of targeted foresight exercises to determine the nature and scope of data preservation activities and the types of data most likely to be needed in the future.

The means to achieve such recommendations even for complex scenarios is to use the principles of Superordinate Knowledge, which integrate arbitrary knowledge over theory and practice. Core assembly elements of Superordinate Knowledge [2] are:

- Methodology.
- Implementation.
- Realisation.

Separation and integration of assemblies have proven beneficial for building solutions with different disciplines, different levels of expertise.

IV. METHODOLOGY OF KNOWLEDGE MAPPING

The methodology can be used for creating new object and entity context environments, e.g., in knowledge mining context. The following steps describe the methodology.

- 1) Start is an arbitrary object.
- 2) Object / entity analysis.
- 3) Object / entity mapping.
- 4) Context creation.
- 5) Result is an object and/or entity with a new context environment.

Objects can be arbitrary objects, unstructured or structured, unreferenced or referenced, e.g., containing different entities of content. The methodology is not limited to any possibly restricted implementation or platform. In case of textual objects and entities, the object can, e.g., be a text document. In case the mapping targets on geo-referencing otherwise non geo-referenced objects or entities, then the mapping can be considered a spatial mapping. With the latter target the context creation can be considered a spatial visualisation.

The methodology of knowledge mapping for arbitrary objects and entities can be schematically summarised (Figure 1).

For example (Figure 2): When the object is a plain text-object and creating spatial visual context is the target, then the steps can be implemented with object and entity analysis, spatial object / entity mapping, and spatial visualisation for creating an object / entity spatial mapping in a new context.

The targets for the case study are spatial visualisation and context. The implementation architecture of mapping arbitrary objects and entities to a new object context environment is shown in Figure 3. Data and modules are provided by Knowledge Resources. The originary resources deliver the data objects and entities, which can be unstructured or structured. The application resources and components contain appropriate modules for the required steps:

- The object is retrieved,
- possible object entities are extracted,
- object data resources are being analysed,
- objects are being compared,
- a conceptual mapping is performed on objects,
- spatial mapping is performed on objects,
- appropriate spatial media is generated,
- including media formats and colourisation, and
- a spatial visualisation is performed.
- The result is an object / entity instance in a new context environment.

The modules and filters perform the analysis and handle the objects and entities, e.g.,

- entities in different context inside an object,
- transcriptions, transliterations, translations,
- abbreviations, acronyms, ...

In many cases, additional handling of data will be desired, even if not essential for the procedure of a method or the operation of a service. For example, in case of textual objects and entities a number of aspects exist, which contribute to the attainment of a certain quality:

- Differently organised or structured entities per object.
- Sub-entities, multiple entities in a pseudo-entity.
- Inconsistencies in data.
- Errors in data.
- Typographic differences.
- Ambiguous or plurivalent entities.
- Multi-lingual entities.
- Different diction.
- Different syntax.
- Different element ordering in entities.
- Different structures.
- Time dependencies of aspects, mapping, and meaning.

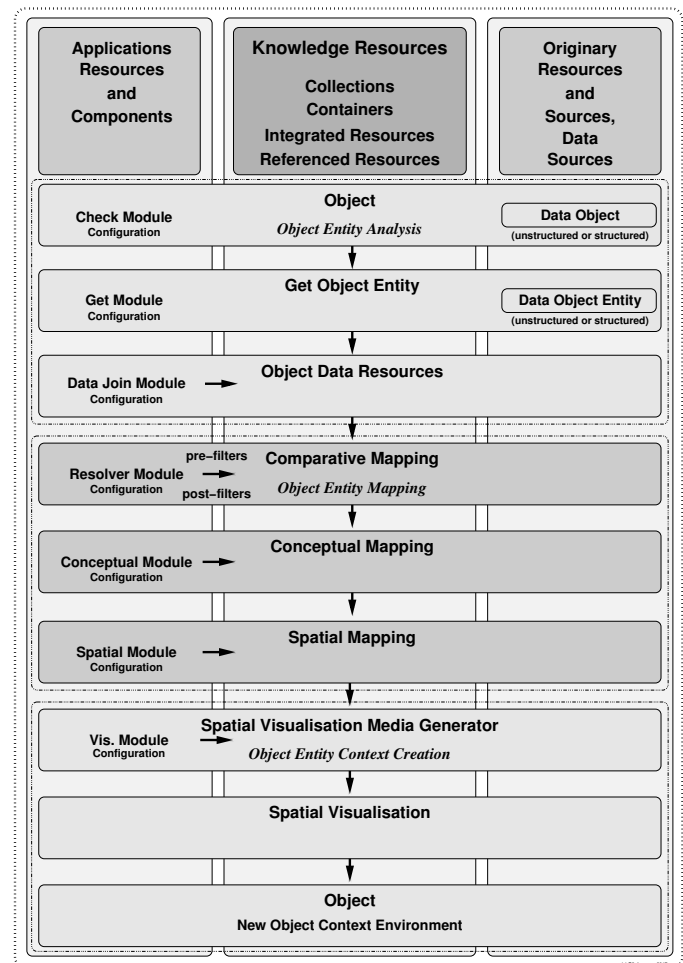


Figure 3. Architecture for implementation of mapping arbitrary objects and entities to new context environments, e.g., spatial visualisation and context.

Data and modules are provided by Knowledge Resources, originary resources, and application resources and components.

- Different character sets.
- Different formatting.

Any of these and comparable aspects are handled by the modules and appropriate pre- and post-filters. With the case study, for the above aspects respective research was conducted gathering various data and developing suitable methods over several years, data which can be deployed to create filters, which were used for holding the results presented here.

It is required to abstract certain information in many application scenarios, e.g., for generalisation or privacy. Besides any kind of filter, the method also allows to implement fuzziness in a flexible and wide range of ways. For example, on the one hand a precise location can be reduced to city, region, or country. Comparable but different locations can be unified to one different location representing a larger area. On the other hand, location coordinates can be automatically or manually reduced in precision and/or equipped with an offset. With these means, workflows can deliver kind of “Fuzzy Context”, e.g., a fuzzy location, providing a precision level of a public region instead of showing a certain building in a result.

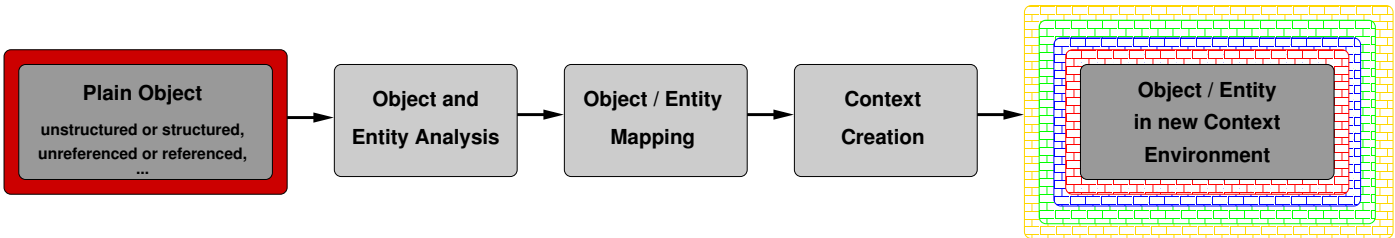


Figure 1. Methodology mapping arbitrary objects and entities for creating new context environments. The methodology requires the major complementary steps of object/entity analysis, mapping, and context creation. Depending of the object, the steps can be implemented using different tools.

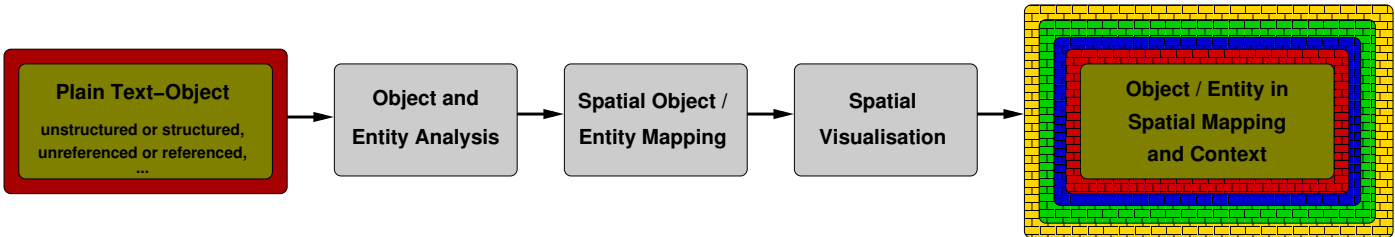


Figure 2. Plain text to spatial mapping context: Mapping arbitrary text objects and entities to new spatial mapping and context. In this case the object is plain text, analysis is conducted with knowledge-mining-in-text algorithms, mapping is spatial mapping, and context creation is spatial visualisation context.

V. PREVIOUS WORK, COMPONENTS, AND RESOURCES

For the implementation of case studies, the modules are built by support of a number of major components and resources, which can be used for a wide range of applications, e.g., creation of resources and extraction of entities.

The facility for consistently describing knowledge is a valuable quality, especially conceptual knowledge, e.g., using the Universal Decimal Classification (UDC). The knowledge resources objects can refer to main UDC-based classes, which for this publication are taken from the Multilingual Universal Decimal Classification Summary (UDCC Publication No. 088) [25] released by the UDC Consortium under the Creative Commons Attribution Share Alike 3.0 license [26] (first release 2009, subsequent update 2012).

UDC provides auxiliary signs [27], shown in Table I, which represent kinds of standardised “operations”.

TABLE I. UDC COMMON AUXILIARY SIGNS USED WITH CLASSIFICATION REFERENCES (ENGLISH VERSION).

Sign	Description (English)
+	Coordination. Addition (plus sign).
/	Consecutive extension (oblique stroke sign).
:	Simple relation (colon sign).
::	Order-fixing (double colon sign).
[]	Subgrouping (square brackets).
*	Introduces non-UDC notation (asterisk).
A/Z	Direct alphabetical specification.

Using these features UDC allows the creation of faceted knowledge. UDC code references based on the main tables of the UDC [28] are shown in Table II. “UDC:” is the designated notation of references for knowledge resources used with references in ongoing projects. The UDC illustrates the width and depth of knowledge dimensions, multi-disciplinary content

and context, and facets. The full details of organisation are available from UDC.

TABLE II. CLASSIFICATION CODE REFERENCES TO UDC MAIN TABLES USED FOR KNOWLEDGE MAPPING (EXCERPT, ENGLISH VERSION).

UDC Code	Description (English)
UDC:0	Science and Knowledge. Organization. Computer Science. Information Science. Documentation. Librarianship. Institutions. Publications
UDC:1	Philosophy. Psychology
UDC:2	Religion. Theology
UDC:3	Social Sciences
UDC:5	Mathematics. Natural Sciences
UDC:6	Applied Sciences. Medicine, Technology
UDC:7	The Arts. Entertainment. Sport
UDC:8	Linguistics. Literature
UDC:9	Geography. Biography. History

Data and objects result from public, commonly available, and specialised Knowledge Resources. The Knowledge Resources are containing factual and conceptual knowledge as well as documentation and instances of procedural and metacognitive knowledge. These resources contain multi-disciplinary and multi-lingual data and context.

The fundament to create mining methods based on this methodology of knowledge mapping is presented with an illustrative scenario. All disciplines, e.g., in the UDC knowledge spectrum, can contribute to this application scenario. Context data for calculations and visualisation also requires cartographic thematic context data. The knowledge resources were integrated with data based on the gridded ETOPO1 1-arc-minute global relief model data [29]. Data can be composed from various sources, e.g., adding Shuttle Radar Topography Mission (SRTM) data [30] from the Consultative Group on International Agricultural Research (CGIAR) [31].

The Network Common Data Form (NetCDF) [32] devel-

oped by the University Corporation for Atmospheric Research (UCAR/Unidata), [33], National Center for Atmospheric Research (NCAR) [34] is used for spatial context data. NetCDF is an array based data structure for storing multi-dimensional data. A NetCDF file is written with an ASCII header and stores the data in a binary format, e.g., with a mapping suite.

The Generic Mapping Tools (GMT) [35] suite application components are used for handling the spatial data, applying the related criteria, and for the visualisation.

The visualisation files generated from the mapping results are using the Keyhole Markup Language (KML), an eXtended Markup Language (XML) based format for specifying spatial data and content. KML is considered an official standard of the Open Geospatial Consortium (OGC). The KML description can be used with many spatial components and purposes, e.g., with a Google Earth or Google Maps presentation [36], with a Marble representation [37], using OpenStreetMap (OSM) [38] and national data, e.g., [39].

Modules are employing Perl Compatible Regular Expressions (PCRE) for specifying common string patterns and Perl [40] for component wrapping purposes with this case study.

VI. MULTI-DIMENSIONAL CONTEXT CREATION

The following sections provide information regarding implemented components (lxlcoord, module for location coordinates) and a practical case study, which was done for demonstrating the methodology of mapping objects and entities, creating new context environments. The case study shows components, which were built for mapping scenarios creating spatial context (Figure 3) and illustrates new insights and relevance for knowledge creation and advanced mining.

A. The components

All the components and modules required for the architecture (Figure 3) were implemented. The following components were created for the practical implementation of the three major central modules, object / entity analysis, mapping, and context creation, demonstrating all steps of the methodology.

- The object / entity analysis modules process objects for entities, which can be fed into a mapping mechanism.
- The pre-filters change, mark, and remove entities before the mapping modules try to create entity mappings.
- The mapping modules do have the task to deliver spatial coordinates for appropriate entities.
- The post-filters change, mark or remove entities after the resolver worked on entities for a spatial mapping.
- The context creation modules deliver the geo-referencing for a spatial application.

The modules can be centralised or distributed, e.g., implemented as a local directory of comparable and resolved entities or an online service. Appropriate directories can be provided by knowledge resources as well as by spatial mapping services.

Change processes in pre- and post-filters can include unification, improvements for resolvability, mapping and so on.

Different application components with different features can be deployed for dynamical and interactive use and visualisation, e.g., GMT, Marble, and Google Maps.

B. Case study: From plain text to spatially linked context

The following passages show some major steps for creating spatially linked context from plain text (Figure 4), which were used in the workflows required for the case studies.

```
1 <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" ... <title>
2 GEOProcessing 2018 ...</title>
3 ..., Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance
4 (HLRN), Germany ...
5 ..., Technion - Israel Institute of Technology, Haifa, Israel<br />
6 ..., Consiglio Nazionale delle Ricerche - Genova, Italy<br />
7 ..., Centre for Research in Geomatics - Laval University, Quebec, Canada <br />
8 ..., Curtin University, Australia<br />
9 ..., Lomonosov Moscow State University, Russia<br />
10 ..., FH Aachen, Germany<p> ...
11 <p>..., Universiti Tun Hussein Onn Malaysia, Malaysia<br />
12 ..., Cardiff University, Wales, UK<br />
13 ..., Universidade Federal do Rio Grande, Brazil<br />
14 ..., GIS unit Kuwait Oil Company, Kuwait<br />
15 ..., Middle East Technical University, Turkey<br />
16 ..., University of Sharjah, UAE<br />
17 ..., Georgia State University, USA<br />
18 ..., Centre for Research in Geomatics - Laval University, Quebec, Canada<br />
19 ..., Environmental Systems Research Institute (ESRI), USA<br />
20 ..., ORT University - Montevideo, Uruguay<br /> ...
```

Figure 4. Mapping target: Single object, unstructured text (excerpt).

The single data object contains mostly unstructured text [41] (status of November 2017), markup, and formatting instructions. Passages not relevant for demonstration were shortened to ellipses. Figure 5 shows the object content after automatically integrated with the Knowledge Resources via a join module.

```
1 GEOProcessing 2018 [...]: ...
2 ..., Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster
3 / North-German Supercomputing Alliance (HLRN), Germany ...
4 ..., Technion - Israel Institute of Technology, Haifa, Israel
5 ..., Consiglio Nazionale delle Ricerche - Genova, Italy
6 ..., Centre for Research in Geomatics - Laval University, Quebec, Canada
7 ..., Curtin University, Australia
8 ..., Lomonosov Moscow State University, Russia
9 ..., FH Aachen, Germany ...
10 ..., Universiti Tun Hussein Onn Malaysia, Malaysia
11 ..., Cardiff University, Wales, UK
12 ..., Universidade Federal do Rio Grande, Brazil
13 ..., GIS unit Kuwait Oil Company, Kuwait
14 ..., Middle East Technical University, Turkey
15 ..., University of Sharjah, UAE
16 ..., Georgia State University, USA
17 ..., Centre for Research in Geomatics - Laval University, Quebec, Canada
18 ..., Environmental Systems Research Institute (ESRI), USA
19 ..., ORT University - Montevideo, Uruguay ...
```

Figure 5. Object instance representation after integration (excerpt).

The Object Entity Mapping can associate relevant objects, e.g., via conceptual knowledge and comparative methods. Table III shows an excerpt of the conceptual data (UDC) used for characteristics and place classification, creating spatial context.

TABLE III. CLASSIFICATION REFERENCES, OBJECT/ENTITY ANALYSIS AND MAPPING: CHARACTERISTICS & PLACE (LX [17]).

UDC Code	Description (English, excerpt)
UDC:(1)	Place and space in general. Localization. Orientation
UDC:(100)	Universal as to place. International. All countries in general
UDC:-05	Common auxiliaries of persons and personal characteristics
UDC:-057.4	Professional or academic workers
UDC:378	Higher education. Universities. Academic study

The codes especially reflect the common auxiliaries of general characteristics and place with the analysis of the object and entities, e.g., affiliation and spatial location.

Figure 6 shows an excerpt with possible entities of locations after an object entity analysis and mapping.

```

1 ...
2 Centre for Research in Geomatics, Laval University, Quebec, Canada
3 Curtin University, Australia
4 Lomonosov Moscow State University, Russia ...
5 Universiti Tun Hussein Onn Malaysia, Malaysia ...
6 Environmental Systems Research Institute (ESRI), USA ...

```

Figure 6. Possible place entities after object / entity analysis (excerpt).

After object entity analysis, filters, and mapping, a resolver module can equip the entities with geo-references (Figure 7).

```

1 ...
2 -71.2747424,46.7817463, Centre for Research in Geomatics, Laval University,
  Quebec, Canada
3 115.8944182,-32.0061951 Curtin University, Australia
4 37.5286696,55.7039349 Moscow State University, Russia ...
5 103.0855782,1.858626,Universiti Tun Hussein Onn Malaysia, Malaysia ...
6 -117.195686,34.056077,Environmental Systems Research Institute (ESRI), USA ...

```

Figure 7. Resolver module result: Resulting entities equipped with geo-references after object entity analysis, filters, and mapping (excerpt).

For this result, the pre- and post filters handled all issues as described. The entries are shown in a special 3 column Comma Separated Value (CSV) format. The GMT format for the geo-referenced CSV is straight forward (Figure 8).

```

1 ...
2 -71.2747424 46.7817463 Centre for Research in Geomatics, Laval University,
  Quebec, Canada
3 115.8944182 -32.0061951 Curtin University, Australia
4 37.5286696 55.7039349 Moscow State University, Russia ...
5 103.0855782 1.858626 Universiti Tun Hussein Onn Malaysia, Malaysia ...
6 -117.195686 34.056077 Environmental Systems Research Institute (ESRI), USA ...

```

Figure 8. Geo-references object entity in GMT format (excerpt).

The context creation includes the media generation. Figure 9 excerpts a KML representation of the above geo-referenced entities, resulting from the original mapping.

```

1 <?xml version="1.0" encoding="UTF-8"?>
2 <kml xmlns="http://www.opengis.net/kml/2.2">
3 <Document>
4 <name>Locations</name>
5 <Folder><name>Conferences</name><style id="locationsconferences"><BalloonStyle>
6 <text>&lt;b&gt;&lt;font color="#0000CC" size="+2"&gt;[name]&lt;/font&gt;&lt;b&gt;&lt;br/&gt;&lt;br/&gt;&lt;
  font face="Courier"&gt;[description]&lt;/font&gt;&lt;br/&gt;&lt;br/&gt;[address]
7 &lt;/id&gt;
8 &lt;[Snippet]
9 &lt;[geDirections]
10 &lt;]]&gt;&lt;text&gt;&lt;/BalloonStyle&gt;
11 &lt;IconStyle&gt;&lt;Icon&gt;&lt;href&gt;http://maps.google.com/mapfiles/kml/pushpin/grn-pushpin.
  png&lt;/href&gt;&lt;/Icon&gt;&lt;/IconStyle&gt;&lt;/style&gt; ...
12 &lt;Placemark&gt;&lt;name&gt;Centre for Research in Geomatics&lt;/name&gt;
13 &lt;description&gt;Centre for Research in Geomatics, Laval University Quebec Canada&lt;/
  description&gt;
14 &lt;styleUrl&gt;#locationsconferences&lt;/styleUrl&gt;
15 &lt;Point&gt;&lt;coordinates&gt;-71.2747424,46.7817463,0&lt;/coordinates&gt;&lt;/Point&gt;&lt;/Placemark&gt;
16 &lt;Placemark&gt;&lt;name&gt;Curtin University&lt;/name&gt;
17 &lt;description&gt;Curtin University, Australia&lt;/description&gt;
18 &lt;styleUrl&gt;#locationsconferences&lt;/styleUrl&gt;
19 &lt;Point&gt;&lt;coordinates&gt;115.8944182,-32.0061951,0&lt;/coordinates&gt;&lt;/Point&gt;&lt;/Placemark&gt;
20 &lt;Placemark&gt;&lt;name&gt;Moscow State University&lt;/name&gt;
21 &lt;description&gt;Moscow State University, Russia&lt;/description&gt;
22 &lt;styleUrl&gt;#locationsconferences&lt;/styleUrl&gt;
23 &lt;Point&gt;&lt;coordinates&gt;37.5286696,55.7039349,0&lt;/coordinates&gt;&lt;/Point&gt;&lt;/Placemark&gt;
24 ...
25 &lt;Placemark&gt;&lt;name&gt;Universiti Tun Hussein Onn Malaysia&lt;/name&gt;
26 &lt;description&gt;Universiti Tun Hussein Onn Malaysia, Malaysia&lt;/description&gt;
27 &lt;styleUrl&gt;#locationsconferences&lt;/styleUrl&gt;
28 &lt;Point&gt;&lt;coordinates&gt;103.0855782,1.858626,0&lt;/coordinates&gt;&lt;/Point&gt;&lt;/Placemark&gt; ...
29 &lt;Placemark&gt;&lt;name&gt;Environmental Systems Research Institute (ESRI)&lt;/name&gt;
30 &lt;description&gt;Environmental Systems Research Institute (ESRI), USA&lt;/description&gt;
31 &lt;styleUrl&gt;#locationsconferences&lt;/styleUrl&gt;
32 &lt;Point&gt;&lt;coordinates&gt;-117.195686,34.056077,0&lt;/coordinates&gt;&lt;/Point&gt;&lt;/Placemark&gt;
33 ...
34 &lt;/Folder&gt;
35 &lt;/Document&gt;
36 &lt;/kml&gt;
</pre>
</div>
<div data-bbox="79 807 471 832" data-label="Caption">
<p>Figure 9. Media representation (KML) of geo-referenced object entities, resulting from original mapping (excerpt).</p>
</div>
<div data-bbox="75 841 493 871" data-label="Text">
<p>A global view of all resulting entities automatically analysed and mapped from the single object [41] is shown in</p>
</div>
<div data-bbox="513 68 942 125" data-label="Text">
<p>Figure 10. The single-object-view integrates the new spatial context of the object entities with a high precision topographic-oceanographic thematic view. The bullets are very much over-sized for this illustration.</p>
</div>
<div data-bbox="513 125 942 250" data-label="Text">
<p>The respective components are provided by GMT suite applications, especially <code>pscoast</code> and <code>gmtselect</code> [17], which allow a multitude of spatial operations and criteria in context with the entities. Further, KML can be used with many spatial applications, e.g., with Marble and Google Maps. Generators can be configured to mark different types of locations with different markers. It is also possible to automatically mark locations with thumbnail photos being associated with the respective location and so on.</p>
</div>
<div data-bbox="513 249 942 332" data-label="Text">
<p>With a general approach, on knowledge side, the universal classification can classify any location and context, the more, it allows to integrate any multi-dimensional context with the full conceptual knowledge. On application component side, GMT provides many functional features, e.g., spatial math, map material assembly, and any map projections.</p>
</div>
<div data-bbox="513 331 942 429" data-label="Text">
<p>Therefore, the integration of just two but very powerful components like UDC and GMT, can provide a huge spectrum of flexibility and fuzziness of expressing location and context. As any documents, e.g., plain texts, Hypertext Markup Language (HTML), and <math>\text{\LaTeX}</math> documents can be handled for affiliation matching can base on fuzzy algorithms based on matching knowledge (e.g., item libraries and classification).</p>
</div>
<div data-bbox="513 428 942 553" data-label="Text">
<p>The knowledge resources and data sets are used for demonstration purposes only and, as far as shown here, are publicly available. The used conceptual knowledge framework is publicly available to the given extent. Further licenses can be obtained, e.g., from the authors of UDC, for further and wider use. Adopters are further free to create their own conceptual knowledge framework to be used with the methodology. The methodology and resulting method frameworks can then be used accordingly.</p>
</div>
<div data-bbox="513 552 942 608" data-label="Text">
<p>When interactivity in the results is a desired target, then components like Marble and Google Earth can provide dynamical features in order to create special cognitive features, e.g., with focus of special details appearing in cognitive context.</p>
</div>
<div data-bbox="513 606 942 704" data-label="Text">
<p>The following implementations show case studies for different context and the resulting output, especially including context of the necessary topography (longitude, latitude, elevation), data, and information used, after the result was visualised via GMT. The examples show automatic spatial mappings of potential POI locations generated for a simple single text object.</p>
</div>
<div data-bbox="523 719 932 733" data-label="Section-Header">
<h2>VII. IMPLEMENTED METHODS: DYNAMICAL CONTEXT</h2>
</div>
<div data-bbox="513 736 942 807" data-label="Text">
<p>Dynamical application components are focussing on targets, which are dynamic. From this point of view, the components themselves are not a matter of change. These components and their contexts are dynamical as the targets are primarily referring to dynamics, to something variable.</p>
</div>
<div data-bbox="513 823 939 839" data-label="Section-Header">
<h3>A. Creating context by knowledge mapping and context data</h3>
</div>
<div data-bbox="513 841 942 871" data-label="Text">
<p>Context data can be any data, which show a reference with the knowledge, with which the case is dealing with. This can</p>
</div>
<div data-bbox="208 968 784 984" data-label="Page-Footer">
<p>2018, © Copyright by authors, Published under agreement with IARIA - www.iaria.org</p>
</div>
```

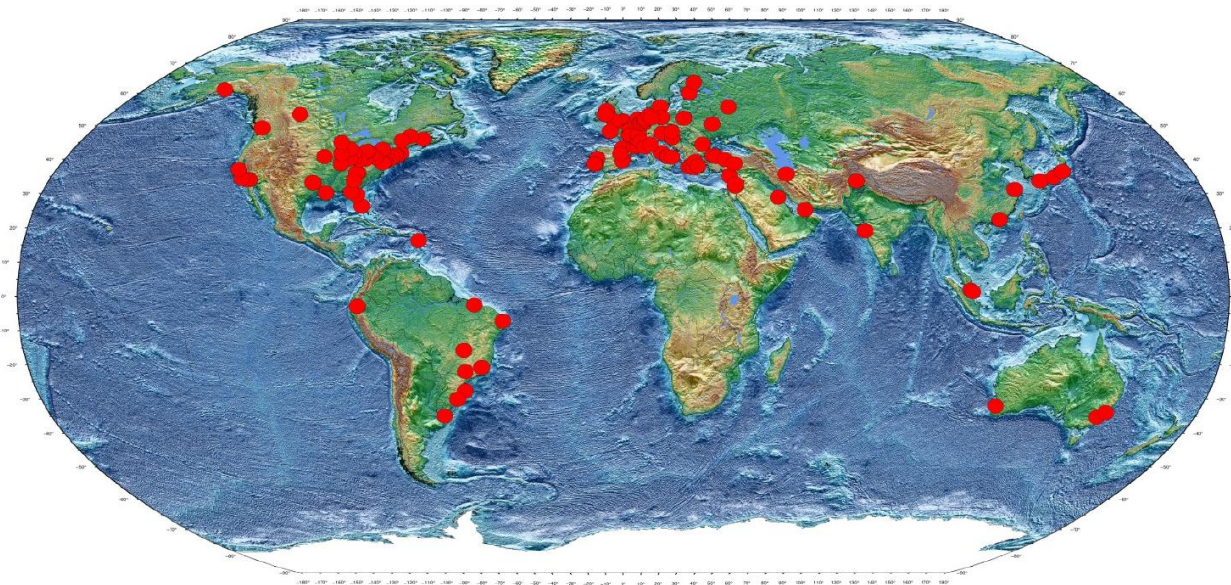


Figure 10. Spatial visualisation result for mapping entities in the text of a single object to a new spatial context and topographic-oceanographic thematic view: Entities resulting from automated analysis and affiliation mapping (red bullets). Sample object: Committee page [41], GEOProcessing 2018 conference, Rome.

be visual context, e.g., concentrations on a thematic mapping, for example with topographic, climatic or political mapping. This can also be mathematical background, e.g., mathematical algorithms being able to describe context and references, e.g., GMT math context or spatial tools context. In case of the above example, such supportive context data can, e.g., be

- climate data,
- weather data,
- political data,
- historical data,
- planetary data (eclipse, areas of visibility, ...),
- routing/navigation data.

Such context, e.g., data allows

- spatial mapping and
- spatial operations.

In consequence, sample facilities are

- location context,
- altitude,
- sea/land/island/...,
- temperature,
- precipitation,
- satellite data context,
- location/POI context.

We can create context-methods for arbitrary knowledge mapping and context strictly based on the Methodology of Knowledge Mapping. The following implemented context-methods directly refer to the above case study and the given knowledge samples.

B. Political context

If the target is a context creation of possible political context of object/entities then the steps are:

- 1) Start: Unstructured object/entities (e.g., from text).

- 2) Analysis of object/entities. Analysis of references, e.g., classifications, concordances, and associations.
- 3) Knowledge Mapping for object/entities. Mapping object/entities with available knowledge.
- 4) Creation of politically referenced context (e.g., political mapping). Referencing with supportive context data (e.g., spatial political maps)
- 5) Result: Object/entities mapped on a dynamical political context map (e.g., Marble, political map).

Figure 11 is a screenshot of an dynamical, interactive view (Marble), a political map context for above created context.



Figure 11. New context for automatically created analysis and mapping of resulting entities of a single object: Political context for labeled entities.

A consecutive mapping allows to analyse the entities in completely new context. For example, parts of an unstructured document can be put into context with any type of n-dimensional information, e.g., historical and climatological context by using spatial information [42] and mapping for finding links. In this

case, data entities can be spatially mapped and associated with multi-dimensional data from many disciplines, and data entities can not only be associated in space but also in time. The data allows to do detailed knowledge mining analysis as well as visual analysis.

The following method implementations create varying contexts and cognitive levels of detail in order to show a small subset of knowledge dimensions for application with spatial targets. Some of the features, e.g., interactive and dynamical features, cannot be illustrated in the figures but will be discussed afterwards.

C. Climate context

If the target is a context creation of possible climate context of object/entities then the steps are:

- 1) Start: Unstructured object/entities (e.g., from text).
- 2) Analysis of object/entities. Analysis of references, e.g., classifications, concordances, and associations.
- 3) Knowledge Mapping for object/entities. Mapping object/entities with available knowledge.
- 4) Creation of climate referenced context (e.g., climatological mapping). Referencing with supportive context data (e.g., spatial climate maps)
- 5) Result: Object/entities mapped on a dynamical climate context map (e.g., Marble, climate map).

For the created context, Figure 12 shows a screenshot of an interactive globe-view (Marble) with climate zone context.

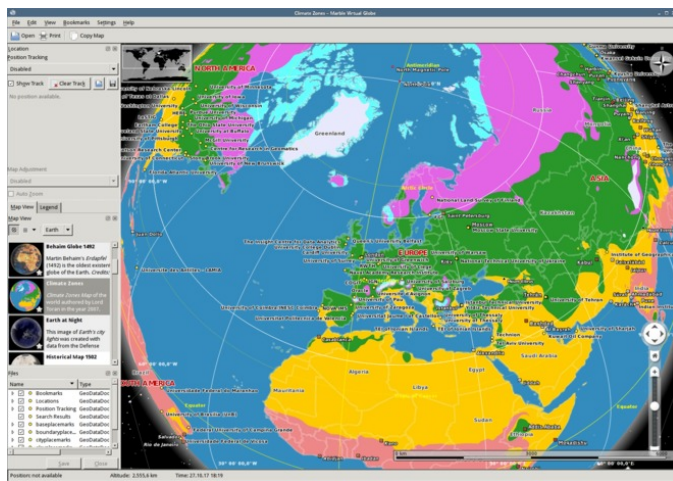


Figure 12. New context for automatically created analysis and mapping of resulting entities of a single object: Climate zones context in 3D.

D. Topographic context

If the target is a context creation of possible global topographic earth context of object/entities then the steps are:

- 1) Start: Unstructured object/entities (e.g., from text).
- 2) Analysis of object/entities. Analysis of references, e.g., classifications, concordances, and associations.
- 3) Knowledge Mapping for object/entities. Mapping object/entities with available knowledge.

- 4) Creation of topographically referenced context (e.g., global topographical mapping). Referencing with supportive context data (e.g., spatial earth topography maps)
- 5) Result: Object/entities mapped on a dynamical topographic context map (e.g., Google Earth, Earth view).

For the created context, Figure 13 shows a screenshot of an interactive view (Google Earth) with Earth view context.

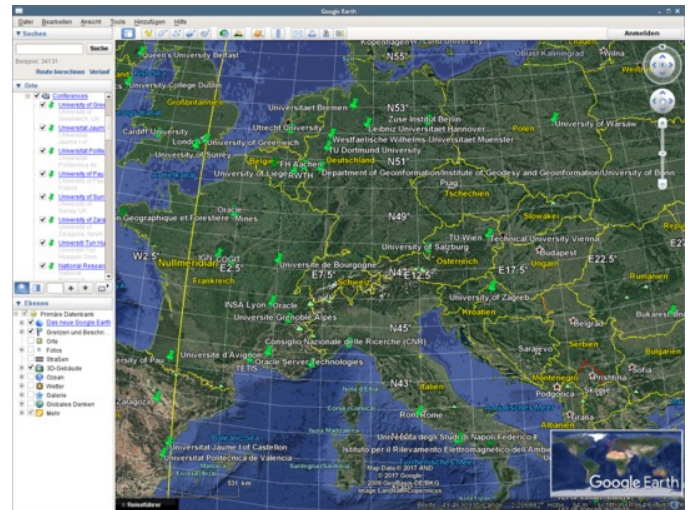


Figure 13. New context for automatically created analysis and mapping of resulting entities of a single object: Google Earth context, labeled entities.

Besides the new context of spatial distribution and according algorithms and math, the new context environments build links in order to associate entities with knowledge from arbitrary disciplines and proceed with further analysis.

Due to conceptual attributes of knowledge mapping and spatial algorithms, the implementation allows high grades of scalability and fuzziness. New context can also be kept and used in learning systems components. This, e.g., can provide conditional object / entity aggregation and time sequences.

E. Super zoom object context

If the target is a context creation of possible super zoom context of object/entities, including context POI and local objects, then the steps are:

- 1) Start: Unstructured object/entities (e.g., from text).
- 2) Analysis of object/entities. Analysis of references, e.g., classifications, concordances, and associations.
- 3) Knowledge Mapping for object/entities. Mapping object/entities with available knowledge.
- 4) Creation of referenced context (e.g., topographical mapping, integration of POI). Referencing with supportive context data (e.g., topic maps)
- 5) Result: Object/entities mapped on a dynamical, zoomable POI context map (e.g., Google Earth).

For the created context, Figure 14 shows a screenshot of an interactive context zoom (Google Earth) of a single entity.

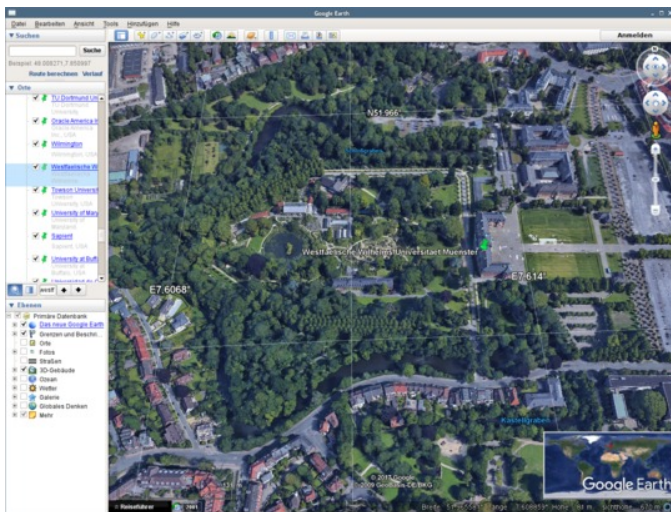


Figure 14. New context for automatically created analysis and mapping of resulting entities of a single object: Google Earth context, super zoom.

This way, any knowledge mapped entity can be made automatically available in its new context. Context can be static and dynamical as well as it can consist of combinations. Many consecutive analysis can be performed as a plethora of algorithms is available to deal with spatial data. Examples are Points of Interest in a certain area or distances to other objects.

F. Public transport context

Figure 15 shows a screenshot of an interactive context view (Marble) of the public transport. The target leading to this application component is the context creation of possible public transport context of object/entities.

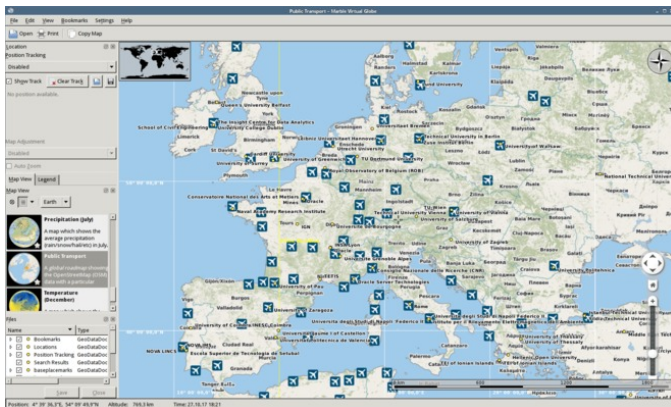


Figure 15. New context for automatically created analysis and mapping of resulting entities of a single object: Public transport.

For this target of context creation of possible public transport context of object/entities the steps are:

- 1) Start: Unstructured object/entities (e.g., from text).
- 2) Analysis of object/entities. Analysis of references, e.g., classifications, concordances, and associations.

- 3) Knowledge Mapping for object/entities. Mapping object/entities with available knowledge.
- 4) Creation of transport referenced context (e.g., public transport mapping). Referencing with supportive context data (e.g., spatial road/rail maps)
- 5) Result: Object/entities mapped on a dynamical transport context map (e.g., Marble).

Questions can be addressed, for example, if there is a correlation between entities referred in an object and the available public transport facilities.

G. Street map context

If the target is a context creation of possible street map context of object/entities then the steps are:

- 1) Start: Unstructured object/entities (e.g., from text).
- 2) Analysis of object/entities. Analysis of references, e.g., classifications, concordances, and associations.
- 3) Knowledge Mapping for object/entities. Mapping object/entities with available knowledge.
- 4) Creation of transport referenced context (e.g., street mapping). Referencing with supportive context data (e.g., spatial road/rail maps)
- 5) Result: Object/entities mapped on a dynamical street context map (e.g., Marble, OSM).

For the created context, Figure 16 shows a screenshot of an interactive context view (Marble, OSM) for Open Street Map.

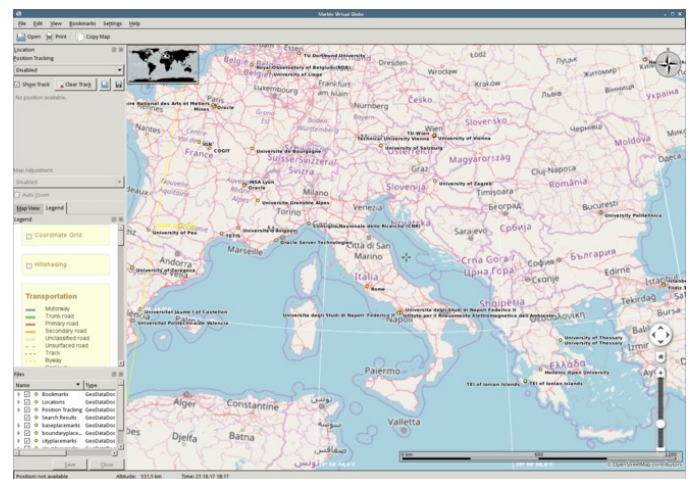


Figure 16. New context for automatically created analysis and mapping of resulting entities of a single object: Open Street Map.

Analysis addressed with navigational context can provide information on relations between entities and their infrastructure based environments.

VIII. DISCUSSION OF MULTI-DIMENSIONAL CONTEXT FEATURES OF METHODOLOGY AND IMPLEMENTATIONS

A. Integration of Context

As the same objects/entities can be brought into different context views, an integration of contexts can deliver new knowledge and insight.

In the above cases, contexts have been created for the entities of a single object, using various dimensions, especially spatial knowledge, time, and other associated knowledge. These contexts, e.g., global distribution, climate, state affiliation, topography, traffic, and navigation in near environment and so on can reveal different insights regarding different object entities.

Even with practical implementations based on publicly available universal conceptual knowledge it is nevertheless only possible to demonstrate a very small range of scenarios of multi-dimensional knowledge context.

Using different spatial framework components, like GMT and Marble, we illustrated some features and what these features can provide for text objects and entities. The methodology allows an overlay and integration of knowledge dimensions. The methods can integrate important data, e.g., raster and vector data, basic spatial object types (esp., points, lines, polygons), and other non spatial data. The implementations allow various types of calculation, e.g., object distance relations and geo-statistics. The methods of analysis range from automated analysis to visual analysis.

A solution integrating GMT can provide practical access to spatial and non-spatial knowledge dimensions, for example:

- Categories of arbitrary land data,
- elevation data, height and depth,
- colourisation of data,
- shading of data,
- borders, political boundaries,
- coastlines,
- water bodies, lakes, permanent rivers, categories of intermittent rivers, categories of canals,
- POI (e.g., cities, sites, locations), and
- context labels.

The individual data can be composed from different sources, different time intervals. Therefore it is possible to compose context from an infinite number of possible combinations.

A solution integrating Marble/KML can provide practical access to spatial and non-spatial knowledge dimensions, for example:

- Categories of arbitrary topic data,
- dynamical changes (e.g., traffic, public transport, POI),
- interactive features,
- environmental and disaster analysis,
- fly-over perspectives,
- time zones,
- lighting (e.g., dawn/dusk areas, daylight/night).

Besides referencing in various dimensions, as illustrated, the methodology allows to create methods for generating time series and animation sequences as well as to handle scalability targets and learning system components. The following features were implemented with the above scenarios.

B. Time series and animation context

All the implementations of methods creating dynamical and statical context, which are based on knowledge mapping –like the above– include features to further create knowledge and knowledge-based implementations on top.

On the one hand, for example, the knowledge and implementations allow to aggregate knowledge, e.g., for creating time series of data. Aggregated knowledge can be used to create secondary and consecutive knowledge and to build and choose appropriate secondary data and applications, e.g., generating KML data and applying it with appropriate application components. The above examples allow animations, e.g., FlyTo views (fly to a certain point of interest) and Tour views (fly to a sequence of points on interests).

On the other hand, the context can be variable over time. This may, for example, be resulting from historical background data like historical satellite data or by integrating real time data.

To this account, aggregated knowledge is a discrete value. For many scenarios, aggregated knowledge otherwise cannot be recreated by other means or at other times. For example, in the above scenario, it is a common case that affiliations are existing for a certain interval in time. Institutions and organisations get founded, and can get renamed, relocated or terminated.

The knowledge to create references is therefore only available in certain time intervals and will change. If creating the same mapping, for example, ten years later may show that references are then missing or having been changed. Aggregating and persistently keeping the knowledge at a time can therefore preserve knowledge, which cannot be created at other times later.

C. Object-entity aggregation

The above object scenario illustrates the multi-dimensional context of one object. The methods can handle arbitrary knowledge objects in any context, e.g., refer to multiple objects and entities in one view, e.g., object instances over an interval of years or different objects within one year, as well as different object instances over an interval of years.

Any precomputed or intermediate result that might be interesting for reuse, either from knowledge or from implementation point of view. There is no general implementation for arbitrary long-term knowledge objects, which means the organisation and management may have to include database components as well as distributed access and interfaces.

Holistic views should also take into consideration that it can be reasonable to also gather and aggregate supportive context data, like old map data, historical POI descriptions and so on.

D. Learning system components

Creating learning system components is just one possible subset within the range of the methodology.

Implementation components and realisations can make use of learning system components for arbitrary scenarios. Learning system components can provide modules in scenarios like the above and reuse knowledge as well as contribute to new knowledge and insight, which allow to collect knowledge developing in terms of time, application, and context.

As an example, learning system components can be built to use aggregated knowledge over long periods of time in order to extend dimensions in knowledge, e.g., in space and time.

Entities in objects may require older context, which cannot be created from present objects, independent of the date of the object. For example, the context of entities in online data does not need to correspond with the date of the hosting file. Lost or missed context may not be possible to be recreated from present data. Long-term application and archiving learning components' results can therefore contribute to a significant quality improvement in many application cases.

E. Checkpointing and knowledge gathering

The methodology allows to create methods, which save checkpointing information for the mapping process.

It can also be reasonable to save resolved references and geolocations persistently because of changes over time, which may not be resolvable later.

When implementing and realising methods, these considerations should be considered with the creation of the modules.

Checkpointing can make use of the mechanisms of knowledge gathering from knowledge as well as from implementation point of view, e.g., collecting time dependent data, data aggregation, and inter module checkpointing data during workflows.

F. Compute resources outline

Depending on the details of resources and scenarios, the computational characteristics can vary. For the case studies some typical numbers are shown in Table IV.

TABLE IV. CASE STUDY COMPUTE AND MAPPING OUTLINE (LX).

Count	Number of
≈10,000,000	Objects addressed in Knowledge Resources
≈100	Corrective patterns for a single entity
≈100	Entities per object
≈100	Entity mappings
≈10,000,000	Comparisons
— 100×100×....	Calculations

As with the implemented methods in the demonstrated case studies it is not uncommon to have millions of comparisons per object and consecutive large numbers of calculations and operations. Multi-dimensional contexts can be much more complex than simple computational frameworks without a knowledge-centric focus. Therefore, a certain computational framework or solution, e.g., parallelisation and linear behaviour related characteristics, can only fit a context to a certain extent.

G. Scalability

The scalability of solutions is most important for any assemblies, the more as the implementations of any methods, which are handling complex context very much correlate with the complexity involved. Due to the complex conditions and dependencies of knowledge, knowledge mapping methods rely very much on the data delivered by knowledge resources. In many cases this is increasingly significant for realisations after an implementation. Realisations mean, for example, building

services based on implementations. In the above case study implementations can handle locations in many ways. Common targets for realisations may be

- precise locations,
- fuzzy locations,
- location precision restricted by criteria, ...

The precision is depending on the associated knowledge and supportive knowledge but also on the intended realisation. The differences of knowledge involved and the consequences on algorithmic and computational efforts may be remarkable.

Within each method, it is possible to take advantage of modifying the contributing factors of scalability when methods are implemented and realised.

IX. CONCLUSION

This paper introduces to multi-dimensional context creation based on the new methodology of Knowledge Mapping and presents the results of the present research. The methodology provides knowledge based mapping of arbitrary objects and entities and creating new multi-dimensional context. The research presented a number of successfully implemented methods, the fundamentals of the theoretical background of the methodology, and the results of the implementation case studies. Analysis and case studies implemented advanced context generation, e.g., with spatial visualisation, 2D, 3D, route mapping, public transport, the prerequisites of context related animation and fly-over tours as well as analysing computational requirements, which are widely scalable, depending on implementation of components and computing architectures.

The methodology fulfills the goals of successfully creating new multi-dimensional context. The Knowledge Mapping methodology can improve complex knowledge mining and associated tasks as well as it can be beneficial for the development of knowledge resources. Practical case studies, evaluated by groups of independent researchers, showed that applying the methodology can create relevant new context for entities in commonly available unstructured data. Regarding the case studies, mapping to spatial context is just one of an arbitrary number of possible mappings, which can be created with the methodology.

The quality of results can significantly benefit from a training and learning phase, depending on context. Here, with resolving nearly all possible place entities with the used new resources, the creation and learning phase of the modules accumulates to several years. The methodology allowed to implement a data-centric checkpointing with the methods. The checkpointing corresponds to associated learning processes.

As shown, in most cases it may be advisable for flexibility to create modular architectures of components instead of monolithic applications. It can further be convenient to consider robustness and reliability of service modules, depending on the architecture of an overall implementation. One means of dealing with infrastructure can be a failure correction, e.g., multiple task runs and check modules.

Future work will concentrate on further developing and improving the mapping modules and features for closer integration with and fostering an even wider range of application of multi-disciplinary knowledge resources.

ACKNOWLEDGEMENTS

We are grateful to the “Knowledge in Motion” (KiM) long-term project, Unabhängiges Deutsches Institut für Multi-disziplinäre Forschung (DIMF), for partially funding this research, implementation, case study, and publication under grants D2017F1P04708 and D2017F1P04812 and to its senior scientific members and members of the permanent commission of the science council, especially to Dr. Friedrich Hülsmann, Gottfried Wilhelm Leibniz Bibliothek (GWLb) Hannover, to Dipl.-Biol. Birgit Gersbeck-Schierholz, Leibniz Universität Hannover, to Dipl.-Ing. Martin Hofmeister, Hannover, and to Olaf Lau, Hannover, Germany, for fruitful discussion, inspiration, practical multi-disciplinary case studies, and the analysis of advanced concepts. We are grateful to Dipl.-Ing. Hans-Günther Müller, Cray, Germany, for his excellent contributions and assistance providing practical private cloud and storage solutions. Our thanks go to all national and international partners in the Geo Exploration and Information cooperations for their constructive and trans-disciplinary support. We are grateful to the Science and High Performance Supercomputing Centre (SHSPC) for long-term support.

REFERENCES

- [1] C.-P. Rückemann, “Methodology of Knowledge Mapping for Arbitrary Objects and Entities: Knowledge Mining and Spatial Representations – Objects in Multi-dimensional Context,” in Proceedings of The Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2018), March 25 – 29, 2018, Rome, Italy. XPS Press, Wilmington, Delaware, USA, 2018, pp. 40–45, rückemann, C.-P. and Doytsher, Y. (eds.), ISSN: 2308-393X, ISBN-13: 978-1-61208-617-0, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2018_3_20_30078 [accessed: 2018-05-19].
- [2] C.-P. Rückemann, “Principles of Superordinate Knowledge: Separation of Methodology, Implementation, and Realisation,” in The Eighth Symposium on Advanced Computation and Information in Natural and Applied Sciences, Proceedings of The 16th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 13–18, 2018, Rhodes, Greece, Proceedings of the American Institute of Physics (AIP). AIP Press, American Institute of Physics, Melville, New York, USA, 2018, ISSN: 0094-243X, (to appear).
- [3] Y. Sun, “Methods for automated concept mapping between medical databases,” *Journal of Biomedical Informatics*, vol. 37, 2004, pp. 162–178, DOI: <https://doi.org/10.1016/j.jbi.2004.03.003> [accessed: 2018-05-19].
- [4] J. Y. Sun and Y. Sun, “A System for Automated Lexical Mapping,” *Journal of the American Medical Informatics Association*, vol. 13, no. 3, 2006, pp. 334–343, DOI: 10.1197/jamia.M1823.
- [5] “Automatic Term Mapping,” 2017, PubMed Tutorial - Building the Search - How It Works, URL: https://www.nlm.nih.gov/bsd/disted/pubmedtutorial/020_040.html [accessed: 2018-05-19].
- [6] N. J. van Eck, L. Waltman, E. C. M. Noyons, and R. K. Buter, “Automatic term identification for bibliometric mapping,” *Scientometrics*, vol. 82, no. 3, 2010, pp. 581–596, URL: <https://link.springer.com/article/10.1007/s11192-010-0173-0> [accessed: 2018-05-19].
- [7] “Automated Mapping Applications,” 2017, interpret Geospatial Solutions, URL: <http://www.interpret.co.nz/projects/automated-mapping-applications/> [accessed: 2018-05-19].
- [8] R. Haga and K. Feigh, “Context maps-classifying contextual influence for decision support system design,” in Proceedings of the Digital Avionics Systems Conference (DASC 2015), Sep. 13–17, 2015, Prague, Czech Republic. IEEE CPS, 2015, ISBN: 978-1-4799-8940-9 (Electronic Proceedings), DOI: 10.1109/DASC.2015.7311576.
- [9] M. Lauruhn and P. Groth, “Sources of Change for Modern Knowledge Organization,” *Systems*, Nov. 2016, URL: <https://arxiv.org/abs/1611.00217> [accessed: 2018-05-19].
- [10] E. König, “Knowledge Organisation Systems in Change (German: Wissensorganisationssysteme im Wandel),” *library essentials*, LE_Informationsdienst, Dec. 2016, 2016, pp. 7–10, ISSN: 2194-0126, URL: <http://www.libess.de> [accessed: 2018-05-19].
- [11] W. He and L. Yang, “Using wikis in team collaboration: A media capability perspective,” *Information & Management*, vol. 53, no. 7, 2016, pp. 846–856, ISSN: 0378-7206, URL: <http://dl.acm.org/citation.cfm?id=3006192> [accessed: 2018-05-19].
- [12] E. König, “On the Efficiency of Wikis as Tools for Collaboration and Knowledge Exploitation (German: Zur Effizienz von Wikis als Tools für Zusammenarbeit und Wissensgewinnung),” *library essentials*, LE_Informationsdienst, Dec. 2016, 2016, pp. 5–7, ISSN: 2194-0126, URL: <http://www.libess.de> [accessed: 2018-05-19].
- [13] R. Barkowski, “WoodApps: improvement in collaboration along the wood value chain through knowledge-based methods and mobile applications, Forschungsbericht, Schlussbericht, Laufzeit des Vorhabens: 01. Nov. 2011 – 31. Dez. 2014, Stand: 17. Feb. 2016,” HCN e.V., Wismar, Deutschland, 2016, Technical Report, 64 Blätter, 1 ungezähltes Blatt, DOI: 10.2314/GBV:868672297, URL: <http://edok01.tib.uni-hannover.de/edoks/e01fb16/868672297.pdf> [accessed: 2018-05-19].
- [14] “Venice Time Machine,” 2018, URL: <http://vtm.epfl.ch/> [accessed: 2018-05-19].
- [15] C.-P. Rückemann, “Computation and Knowledge Mapping for Data Entities,” in Proceedings of The Eighth International Conference on Advanced Communications and Computation (INFOCOMP 2018), July 22–26, 2018, Barcelona, Spain. XPS Press, Wilmington, Delaware, USA, 2018, pp. 7–12, ISSN: 2308-3484, ISBN: 978-1-61208-655-2, URL: http://www.thinkmind.org/index.php?view=article&articleid=infocomp_2018_1_20_60031 [accessed: 2018-11-25].
- [16] C.-P. Rückemann, “Progressive Advancement of Knowledge Resources and Mining: Integrating Content Factor and Comparative Analysis Methods for Dynamical Classification and Concordances,” *International Journal on Advances in Systems and Measurements*, vol. 11, no. 1&2, 2018, pp. 47–60, ISSN: 1942-261x, URL: http://www.thinkmind.org/index.php?view=article&articleid=sysmea_v11_n12_2018_5/ [accessed: 2018-11-25].
- [17] C.-P. Rückemann, “Methodology and Integrated Knowledge for Complex Knowledge Mining: Natural Sciences and Archaeology Case Study Results,” in Proceedings of The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2017), March 19 – 23, 2017, Nice, France. XPS, 2017, pp. 103–109, Rückemann, C.-P. and Doytsher, Y. and Xia, J. C. and Braz, F. J. (eds.), ISSN: 2308-393X, ISBN-13: 978-1-61208-539-5, URL: http://www.thinkmind.org/index.php?view=article&articleid=geoprocessing_2017_7_10_30036 [accessed: 2018-05-19].
- [18] L. W. Anderson and D. R. Krathwohl, Eds., *A Taxonomy for Learning, Teaching, and Assessing: A Revision of Bloom’s Taxonomy of Educational Objectives*. Allyn & Bacon, Boston, MA (Pearson Education Group), USA, 2001, ISBN-13: 978-0801319037.
- [19] C.-P. Rückemann, O. O. Iakushkin, B. Gersbeck-Schierholz, F. Hülsmann, L. Schubert, and O. Lau, “Best Practice and Definitions of Data Sciences – Beyond Statistics,” 2017, Delegates’ Summit, The Seventh Symposium on Advanced Computation and Information in Natural and Applied Sciences (SACINAS), The 15th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 25–30, 2017, Thessaloniki, Greece, URL: http://www.user.uni-hannover.de/cpr/x/publ/2017/delegatessummit2017/rueckemann_icnaam2017_summit_summary.pdf [accessed: 2018-05-19], DOI: <https://doi.org/10.15488/3411> [accessed: 2018-05-19].
- [20] Aristotle, *Nicomachean Ethics*, Volume 1, 2009, Project Gutenberg, eBook, eBook-No.: 28626, Release Date: April 27, 2009, Digitised Version of the Original Publication, Produced by Sophia Canoni, Book provided by Iason Konstantinidis, Translator: Kyriakos Zambas, URL: <http://www.gutenberg.org/ebooks/12699> [accessed: 2018-07-08].

- [21] Aristotle, The Ethics of Aristotle, 2005, Project Gutenberg, eBook, EBook-No.: 8438, Release Date: July, 2005, Digitised Version of the Original Publication, Produced by Ted Garvin, David Widger, and the DP Team, Edition 10, URL: <http://www.gutenberg.org/ebooks/8438> [accessed: 2018-07-08].
- [22] C.-P. Rückemann, F. Hülsmann, B. Gersbeck-Schierholz, P. Skurowski, and M. Staniszewski, Knowledge and Computing. Post-Summit Results, Delegates' Summit: Best Practice and Definitions of Knowledge and Computing, September 23, 2015, The Fifth Symposium on Advanced Computation and Information in Natural and Applied Sciences, The 13th International Conference of Numerical Analysis and Applied Mathematics (ICNAAM), September 23–29, 2015, Rhodes, Greece, 2015, DOI: 10.15488/3409, URL: http://www.user.uni-hannover.de/cpr/x/publ/2015/delegatessummit2015/rueckemann_icnaam2015_summit_summary.pdf [accessed: 2018-07-08], URL: <https://www.tib.eu/en/search/id/datacite%3Adoi~10.15488%252F3409/Best-Practice-and-Definitions-of-Knowledge-and-Computing/> [accessed: 2018-07-08].
- [23] C.-P. Rückemann and F. Hülsmann, "Significant Differences: Methodologies and Applications," "Significant Differences: Methodologies and Applications", KiMrise, Knowledge in Motion Meeting, November 27, 2017, Knowledge in Motion, Hannover, Germany, 2017.
- [24] Organisation for Economic Co-operation and Development (OECD), "OECD Principles and Guidelines for Access to Research Data from Public Funding," 2007, URL: <https://www.oecd.org/sti/sci-tech/38500813.pdf> [accessed: 2018-05-19].
- [25] "Multilingual Universal Decimal Classification Summary," 2012, UDC Consortium, 2012, Web resource, v. 1.1. The Hague: UDC Consortium (UDCC Publication No. 088), URL: <http://www.udcc.org/udccsummary/php/index.php> [accessed: 2018-05-19].
- [26] "Creative Commons Attribution Share Alike 3.0 license," 2012, URL: <http://creativecommons.org/licenses/by-sa/3.0/> [accessed: 2018-05-19].
- [27] "UDC, Common Auxiliary Signs," 2018, Universal Decimal Classification (UDC), URL: <https://udcdata.info/078885> [accessed: 2018-10-14].
- [28] "UDC Summary Linked Data, Main Tables," 2018, Universal Decimal Classification (UDC), URL: <https://udcdata.info/078887> [accessed: 2018-10-14].
- [29] C. Amante and B. W. Eakins, "ETOPO1 1 Arc-Minute Global Relief Model: Procedures, Data Sources and Analysis," 2009, NOAA Technical Memorandum NESDIS NGDC-24. National Geophysical Data Center, NOAA. DOI: 10.7289/V5C8276M [accessed: 2018-05-19], World Data Service for Geophysics, Boulder, USA, National Geophysical Data Center, National Centers for Environmental Information (NCEI), National Oceanic and Atmospheric Administration (NOAA), URL: <http://www.ngdc.noaa.gov/mgg/global/relief/ETOPO1/data/> [accessed: 2018-05-19].
- [30] "CGIAR Consortium for Spatial Information (CGIAR-CSI)," 2018, URL: <http://www.cgiar-csi.org> [accessed: 2018-05-19].
- [31] "Consultative Group on International Agricultural Research (CGIAR)," 2018, URL: <http://www.cgiar.org> [accessed: 2018-05-19].
- [32] "NetCDF – Network Common Data Form," 2018, DOI: <http://doi.org/10.5065/D6H70CW6> [accessed: 2018-05-19] URL: <http://www.unidata.ucar.edu/software/netcdf/> [accessed: 2018-05-19].
- [33] "University Corporation for Atmospheric Research (UCAR)," 2018, URL: <https://ncar.ucar.edu/> [accessed: 2018-05-19].
- [34] "National Center for Atmospheric Research (NCAR)," 2018, URL: <https://www.ucar.edu/> [accessed: 2018-05-19].
- [35] "GMT - Generic Mapping Tools," 2018, URL: <http://gmt.soest.hawaii.edu/> [accessed: 2018-05-19].
- [36] "Google Maps," 2018, URL: <http://www.google.com/maps> [accessed: 2018-05-19].
- [37] "Marble," 2018, URL: <https://marble.kde.org/> [accessed: 2018-05-19].
- [38] "OpenStreetMap (OSM)," 2018, URL: <http://www.openstreetmap.org> [accessed: 2018-05-19].
- [39] "OpenStreetMap (OSM) - Deutschland," 2018, URL: <http://www.openstreetmap.de> [accessed: 2018-05-19].
- [40] "The Perl Programming Language," 2018, URL: <https://www.perl.org/> [accessed: 2018-05-19].
- [41] "GEOProcessing 2018: Committees," 2018, the Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services (GEOProcessing 2018) March 25–29, 2018 – Rome, Italy, URL: <https://www.iaria.org/conferences2018/ComGEOProcessing18.html> [accessed: 2018-05-19].
- [42] "Marble Maps," 2018, URL: <https://marble.kde.org/maps.php> [accessed: 2018-05-19].

An International Survey of Practitioners' Views on Personas:

Benefits, Resource Demands and Pitfalls

Engie Bashir

Computer Engineering and Informatics
Middlesex University Dubai
Dubai, United Arab of Emirates
e-mail: e.bashir@mdx.ac.ae

Simon Attfield

Department of Computer Science
Middlesex University
London, United Kingdom
e-mail: s.attfield@mdx.ac.uk

Abstract— Opinions differ on the relative costs and benefits of personas as used in Interaction design (IxD) and User Experience (UX). And yet there has been little research to systematically elicit attitudes towards them held by IxD and UX professionals. We report a 'state-of-practice' survey conducted with IxD/UX professionals called 'What's Hot in Interaction Design' and focus here on 20 items from the survey that elicited usage of and opinions about personas. The survey items were derived from opinions and reports collated from the academic and professional literature. We use factor analysis to reduce the items to a fundamental set of areas or concerns (factors), and use significance testing to test for agreement on each item including an analysis of the strength of opinion using odds ratio. According to the findings, 64% of respondents use personas with usage during research, design & evaluation phases. The factor analysis shows that opinions fall into three broad areas: benefits, resource demands, and pitfalls. Practitioners tend to agree that personas have a range of benefits, but that they make demands on specific kinds of resources and there are some specific pitfalls—all of which we report. We discuss implications for improving personas through enhanced methods and tools, and curricula.

Keywords—*persona usage; persona factors; prioritising persona attitudes; interaction design; theory and practice; tools and curricula.*

I. INTRODUCTION

This paper presents findings from a survey of Interaction Design (IxD) and User Experience (UX) professionals concerning their views about personas, presented initially in [1] and extended using exploratory factor analysis, which reveals three factors: benefits, resource demands and pitfalls. In 1999, Cooper [2] introduced the idea of personas as a way of anchoring design within a vision of intended users. A persona is a kind of user-model—a *composite archetype* [3] drawn from behavioral data from users of an existing or intended digital product. A set of personas can be created where each represents a group of users with similar behaviors, attitudes, aptitudes, and needs. Methods for creating personas have been suggested by Cooper [3], Pruitt and Adlin [4], and Nielsen [4][5] with semi-automated methods also being proposed [7]–[9].

Personas can have a role in the three phases of the User Centered Design (UCD): User Research & Requirements, Designing & Prototyping and Evaluation. Advocates argue that they promote empathy and help focus design on the goals and characteristics of users. Despite the enthusiasm that some

hold for personas, however, concerns have been raised about issues such as the resources required to create them [4][10]–[13] and their value to the design process [12]–[16].

A review of practitioners' attitudes towards personas via a selection of articles on professional websites revealed views ranging from strong advocacy to skepticism. Although it has been 18 years since the publication of *The Inmates are Running the Asylum* [2], there has been little research to systematically elicit attitudes towards personas held by the people who might use them—Interaction Design and User Experience professionals.

We conducted an online survey called 'What's Hot in Interaction Design' to elicit details of current practices and attitudes of industry professionals. The survey spanned many topics, of which personas was one. Our motivation was to provide stimulus for considering new methods and tools, to inform university syllabus development, and simply to record and report current trends. The survey was in two parts: (1) an initial part about Interaction Design/User Experience practice in general ('main survey'), which included 4 questions about personas, and (2) an optional additional part ('persona survey'), with 16 items (hereafter referred to as 'A1' to 'A16' see Table I) focused on personas. Items were derived from a review of issues raised in the academic literature. The main survey was completed by 173 practitioners. 76 practitioners went on to complete the persona survey.

In this paper, we report results relating to persona use from both the main survey and the persona survey. We also report an exploratory factor analysis that was used to organize the results in terms of a set of more abstract, latent variables. This provides an organizing principle for attitudes towards personas reducing them into to a set of more fundamental factors. We also report an analysis of each item using significance testing and prioritize items using effect size (odds ratio) as a measure of relative strength of feeling.

In Section II, we review background literature that provided the basis for the persona survey items. In Section III, we discuss the survey and analysis method, and in Section IV we report the findings. In the final section, we summarize the results and discuss implications of our findings for interaction design practice.

II. LITERATURE REVIEW

A. Overview on Personas

Cooper introduced the idea of personas in 1999 [2]. Although a method for creating personas was not clearly

articulated at that point, the idea attracted a good deal of attention. According to Cooper, personas offer a balance between formality and informality that carries more nuance than diagrammatic models through capturing users' goals, tasks, characteristics, and environments. The belief was that they could allow design teams from different disciplines and stakeholders to communicate about and empathize with the users and develop more focused designs. Methods for creating personas were subsequently offered that provided a structured approach to the development of personas. These included Pruitt, Grudin and Adlins' 'role-based perspective' [4][10]; Cooper, Reimann and Cronin's 'goal-directed perspective' [2]; and Nielsen's 'engaging perspective' [5]. Cooper, Reimann and Cronin's [2] method is a 7-step approach representing user-goals and including activities, attitudes, aptitudes, motivations, and skills towards a product. Pruitt, Grudin, and Adlin [4][10] agreed on the benefits of personas suggested earlier, but proposed personas as a complementary tool. Their method is a 5-step approach that looks into massive data and attempts to verify the quality and adequacy of persona representation. Nielsen [4][5], who observed variations in persona use, criticized some practitioners for failing to fully appreciate the potential of personas and for adopting marketing archetypes as personas. She offered the 'Engaging Persona' process, which is a 10-step approach aimed at establishing common ground on gathering data related to user needs, attitudes and aptitudes and includes details such as social background, psychological characteristics, and emotional relationship to invoke empathy and avoid stereotyping [17]. The method also included some steps that focus on how to make personas accepted and used by team members.

B. Studies on Personas

Some studies have explored experiences and outcomes of persona creation and use. Blomquist and Arvola [14], for example, observed a design team's first experience with personas. Methods for creating personas were relatively under-developed at that time and the authors found designers lacked confidence in using them for communication or design, concluding a need for expertise and integrating personas within existing knowledge and practice. Chang *et al.* [18] reported a small study with practitioners comparing attitudes of some who used personas and some of who didn't. The study found more positive attitudes towards personas from those who use personas who found it an essential tool for design. The study also found practitioners experimenting with new approaches. Later, Miaskiewicz and Kozar [19] elicited perceived benefits of personas from 19 experts (practitioners who created and used them) and derived a ranked list of 22 benefits, including: providing audience focus, helping to guide decisions, supporting collaboration, acting as a communication aid and guiding evaluation. Mathews *et al.* [16] reported a study of 14 practitioners and observed that those trained on Cooper's method tended to champion personas, whereas those trained in Engineering and Computer Science were 'moderate' persona users, and those trained in HCI and Design were pessimistic. The study also indicated

benefits of personas in helping understand users' needs and context and establishing common ground.

A number of literature sources draw attention to the cost implications of personas creation. LeRouge [20] argued that despite their cost implications, when personas are successfully integrated into a design process by trained team members, the benefits outweigh the costs. Billestrup *et al.* [21] designed a questionnaire survey to investigate the knowledge and use of personas across 60 software development companies within a specific geographical region. The results revealed that more than half of the respondents had not heard of personas while the other respondents stated that personas were not well integrated into the development process. In addition, some problems related to time and budget constraints, limited knowledge with persona methods and inadequacy/shallowness of persona descriptions were reported.

Based on an observational study of design team conversations, Friess [12] questioned the benefits of personas as a tool for communication. Fries' study showed that despite time and resources spent on developing and refining personas, they were only referred to briefly in designers' conversations. Fries, however, resists the conclusion that personas are not useful with the observation that members of the design team who created personas invoked them in conversations much more often than other team members and stakeholders. Tharon [13] commented on the result that, "Leaving the development of the personas to a select few on the team seems likely to ensure that those few are the only members of your team who will benefit from the time and money invested in the personas development."

C. Personas and Empirical Methods of User Research

It is argued across the several methods of creating personas [3]–[5][7]–[9][11] that personas should be derived from user research. The approach suggested by Cooper [2][3] was solely qualitative, involving informal manual clustering of users (based on 'behavioral variables'). Such an approach has raised questions about possibilities of exploiting quantitative data [4][9]–[11], as well as issues of sample size [4][7][9]–[11][15], adequacy of personas in terms of validity and human bias [8][9][11][15], and time and budget implications [4][7][9]–[11][13][15][16][22]. In response to such issues, some have proposed the integration of quantitative research and/or automating clustering methods.

Pruitt, Adlin, and Grudin [4][10] were the first to combine quantitative and qualitative methods based on existing data about users. Their clustering method remained manual and was performed by experts in user research. They suggested validating personas through "sanity checks" and "foundation documents" to link them with the original gathered data. Later, Chapman's and Milham's [15] discussed the unexplored limitations of the former persona methods in terms of significance, accuracy, validity, human bias, and relation to the design of the product. These authors focused on bringing some automation to the process to increase objectivity, improve validity by increasing sample size, whilst also improving the efficiency of the method and making it less dependent on research expertise.

Mulder and Yaar [11] proposed a mixed method for web design personas starting with a quantitative analysis of large-scale market research and website log data and using semi-automated clustering techniques to create market segmentation/user profiles, followed by qualitative analysis such as interviews, field studies or usability tests. Following this, McGinn and Kotamraju [9] suggested designing a survey with agreed attributes to collect large-samples of customer data. *Factor Analysis* (FA) was used from the initial groupings, followed by interviews with selected users to reveal the goals and motivation and to validate group membership.

Maikenzie *et al.* [8] proposed *Latent Semantic Analysis* (LSA) for semi-automated clustering of qualitative interview transcripts data, proposing this method to be “more efficient, less subjective, and less reliant on specialized skills”. Brikey, Walczak, and Burgess [7] reported a study that classified the methods of creating personas in terms of *manual qualitative techniques and semi-automated techniques* (LSA, FA, principal component analysis (PCA) and multivariate cluster analysis (CA)). The findings indicated that LSA semi-automated method, when compared to the manual qualitative method, is not affected by the quantity of data, requires less expertise in clustering, is faster and cheaper, and minimizes human bias. The study also showed that the three automated clustering techniques didn't agree with the cluster assignment done by experts.

In her 10-step approach, Nielsen [5] applies quantitative and qualitative research methods and considers manual clustering techniques (affinity diagrams and empathy maps) to be performed by qualified team members. These approaches each in its own capacity have exploited at least one of the following: sample size, adequacy of persona, time and budget; yet all of them need the expertise in quantitative/qualitative data analysis for clustering users.

D. Professional Literature

We also conducted a review of professional magazines and association websites for articles on personas. Here, mixed opinions can be found along with specific concerns that in many cases echo those expressed in the academic literature. For example, Sholmo [23][24] remarks, “For every designer who uses personas, I have found even more who strongly oppose the technique.” He reflects on his own conversion from negative attitude to positive once he started to develop and use personas “properly” in his work. He attempts to convince detractors to change their perceptions and promotes the use of personas for those who are unfamiliar with the process. Similarly Kellingley [25], another advocate for the development of personas, agreed with many of the criticisms under three headings: “Personas are time-consuming”, “Personas are expensive”, and “Personas need time to show ROI”. However, he argues that more time and money would be spent on building and rebuilding products without considering user requirements and personas. Accordingly, the attempt to reduce cost and time by cutting back on user research and abandoning the use of personas does not hold. In the same way, Bryan [26] discusses three reasons that lead some peers to adverse personas as a design tool. First, the use

of “Analytics”, which he argues can reveal many insights about the design components based on users’ interactions, overlooks how UX designers work and merely specifies user behaviors, which is essential to the UX strategy. Second, A/B and multivariate testing assesses alternative designs in terms of quantitative results, but do not suggest how to reach the best design. Third, in an agile environment UX practitioners feel a burden when creating and designing personas because of time constraints, which again reveals that there is a need for better ways of fitting personas in the UX process.

III. METHOD

A. Survey Design

The main survey contained 29 questions distributed across sections on: (1) demographics; (2) user research; (3) design and prototyping; (4) product development; and (5) evaluation. Section (1) contained questions about respondents’ professional experience, the answers to which determined subsequent sections they were asked to complete. There were four questions about personas in the main survey across the remaining sections.

The persona survey contained 16 items. Each elicited agreement with a series of propositions on a five-point Likert scale. Each proposition represents a possible attitude towards personas. These were derived from the literature by collating opinions and making observations of work reported by relevant academic sources (most appear in the literature review above) and a selection of industry blogs. The propositions are itemized in Table I and each is mapped against its sources.

The studies we used to generate the propositions were typically qualitative and/or longitudinal and based on a small sample drawn from a specific context. In this sense, the survey can be seen as corroborating their findings against a larger and more widely drawn sample. In some cases, sources contradicted each other. Here the survey can be seen as helping to resolve such conflicts. Thus, we believed we converged on a set of concerns that were relevant and might be profitably tested with reference to the experience of a larger sample of practitioners.

It is not uncommon for surveys to use both forward and reverse-keyed versions of items to control for possible acquiescence bias. However, Sonderen *et al.* [28] and Schriesheim & Hill [29] argue that there is little empirical evidence to support this recommendation and demonstrate that it can increase respondent confusion and introduce difficulties in interpretation. Since reverse-keying effectively doubles the size of a survey, which would have a negatively effect on sample size, it was not used here. Hence, we opted for one item per proposition.

B. Participants and recruitment

The target population for the survey was UX/IxD practitioners. Respondents were recruited by non-probabilistic convenience sampling via invitations to online interest groups, and by snowball sampling via the researchers’ professional networks. The requirement of working as a UX/IxD practitioner was included in invitations. Respondents

were asked to give job titles as part of the survey and these were subsequently reviewed for relevance prior to analysis.

TABLE I. THE 16 STATEMENTS USED AS ATTITUDINAL MEASURES TOWARDS PERSONAS. (SUPPORTIVE/UN-SUPPORTIVE INDICATES OPINION ELICITED FROM OR OBSERVATION MADE OF REPORTED WORK)

A1: Personas are time consuming to create/use. Supportive: [4][9]–[11][13][20][21]. Unsupportive: [7][8]
A2: Personas are expensive to create/use. Supportive: [4][9]–[11][20][21] Unsupportive: [7][8]
A3: Representative personas require a lot of data. Supportive [4][9]–[11][20]. Unsupportive: [18][27]
A4: Personas require expertise in qualitative research to create. Supportive: [3][4][10][11][14][18][20]. Unsupportive: [7][8]
A5: Personas require training in persona methods. Supportive: [3]–[5][9]–[11][16][20]
A6: Collaborating around personas is difficult. Supportive: [4][10][14]. Unsupportive: [15]
A7: Personas are often not properly used by teams. Supportive: [12][13]. Unsupportive: [4][8][14]
A8: Personas often represent extreme archetypes Supportive: [3][11][22]
A9: Personas often lack important information related to goals, needs, behaviors, and attitudes. Supportive: [21]. Unsupportive: [2]–[6][10][11]
A10: Persona sets often incorporate redundancy (multiple personas referring to the same characteristics). Supportive: [3][4]
A11: Personas are helpful for understanding users' needs and context. Supportive: [2]–[7][10][11][20][22]. Unsupportive: [14]
A12: Personas are helpful for making design decisions. Supportive: [3][4][6][8][10][11][22]. Unsupportive: [14]–[16]
A13: Personas are helpful for implementing and building Supportive: [3][4][6][10][11][20][22]
A14: Personas are helpful for evaluation. Supportive: [11][19][20][22]. Unsupportive: [21]
A15: Personas are helpful for communicating with stakeholders and team members. Supportive: [2]–[6][8][10][11][20][22]. Unsupportive: [12]–[14]
A16: The personas I use are usually well formed and adequate. Unsupportive: [21]

C. Data Analysis

Responses to each Likert item were coded on a scale of 1 to 5 where 1 = *strongly disagree*, 2 = *disagree*, 3 = *neutral*, 4 = *agree*, 5 = *strongly agree*. For each item, a lower bound one-sample, one-tailed sign test was performed to assess agreement according to the following hypotheses:

H0: The population median response is equal to or less than 'neutral' ($\eta \leq 3$) (i.e., non-agreement)

H1: The population median response is greater than 'neutral' ($\eta > 3$) (i.e., agreement)

Given the multiple tests, Benjamini and Hochberg [30] was used to control for inflated type I error rate ($\alpha_{\text{adjusted}} = .040625$). The odds ratio (OR) (an unstandardized effect size statistic) was also computed for each item and used to organize the responses in terms of strength of expressed opinion.

IV. FINDINGS

A. Demographics

The main survey and the persona survey were completed by 173 and 76 practitioners respectively, with the following self-reported demographics (number in main survey/number in persona survey):

- Job Titles: UX Designers (52/21), UX Researchers (27/13), Senior User Experience Designers (23/13), User Interface Designer / Information Architect (7/2), and others (64/27);
- Years of experience: > 5 yrs (79/36), 3 to 5 yrs (45/17), 1 to 2 yrs (26/11), < 1 year (23/12);
- Countries: UK (56/30), USA (35/13), Sweden (12/7), India (11/2), Norway (8/3), UAE (8/3) and 43/18 others;
- Organization size: 20-99 employees (34/14), 1000-4999 employees (31/13), 10000+ employees (24/13), 100 to 499 employees (24/6), 5000-9999 employees (20/8), 1 to 4 employees (12/3), 10 to 19 employees (9/5), 500 to 999 employees (8/8), 5 to 9 employees (6/6).

Respondents worked with digital products in the areas: websites (134/63); mobile solutions/applications (121/52); consumer technology (73/35); enterprise solutions (67/33); accessibility (62/24); visualization of big data (44/25); smart objects/devices (IOT)(31/10); tabletops/multi-touch surfaces (24/8); wearable technology (19/5); Robotics & AI (16/4); A/R (14/3); VR (11/2); others (35/14).

B. Persona Use

Of the 173 practitioners who completed the main survey 111 (64%) reported using personas in some capacity. Of 105 respondents involved in user research, 78 (74%) reported using personas to represent/communicate user needs based on research studies. Of 109 respondents involved in design and prototyping, 69 (63%) reported using personas for motivating design ideas/decisions and 44 (40%) reported using persona-based inspection for creating/refining the concepts of design. Of 113 respondents involved in evaluation, 34 (30%) reported using persona-based inspection methods.

C. Results from the Personas Survey

Of the practitioners who completed the 'What's Hot in Interaction Design?' survey, 76 went on to complete the optional persona survey.

1) Exploratory Factor Analysis

We wanted to understand the results of the persona survey in terms of a reduced set of underlying factors to use as an organizing principle for attitudes towards personas.

TABLE II. FACTOR LOADINGS AND COMMUNALITIES BASED ON A PAF WITH OBLIMIN ROTATION FOR THE RESPONSE ITEM

	<i>Persona benefits</i>	<i>Persona resources</i>	<i>Persona pitfalls</i>	Communalities
A12 - Personas are helpful for making design decisions.	0.818			0.662
A11 - Personas are helpful for understanding users' needs and context.	0.757			0.587
A13 - Personas are helpful for implementing and building.	0.708			0.507
A14 - Personas are helpful for evaluation.	0.565			0.382
A15 - Personas are helpful for communicating with stakeholders and team members.	0.457			0.245
A4 - Personas require expertise in qualitative research to create.		0.832		0.64
A2- Personas are expensive to create/use.		0.605		0.462
A3 - Representative personas require a lot of data.		0.599		0.424
A5- Personas require training in persona methods.		0.583		0.336
A6- Collaborating around personas is difficult.	-0.376	0.418		0.456
A10 - Persona sets often incorporate redundancy (multiple personas referring to the same characteristics).			0.956	0.781
A8 - Personas often represent extreme archetypes.			0.555	0.322
A9 - Personas often lack important information related to goals, needs, behaviors, and attitudes.			0.501	0.413
A7 - Personas are often not properly used by teams.			0.399	0.271
A1- Personas are time consuming to create/ use.		0.392	0.385	0.474

We used an EFA with Principal Axis Factoring (PAF) since PAF holds no assumption about the distribution of the data. The following steps were followed to ensure that validity and reliability of the final solution.

The survey data fulfilled the following suitability criteria:

1. Data on each item showed a correlation of at least 0.3 with at least one other item;
2. The Kaiser-Meyer-Olkin measure was 0.746 (greater than 0.6), and Bartlett's test of sphericity was significant ($\chi^2(120) = 435.131, p < .001$);
3. The anti-image correlation matrix diagonals were all greater than 0.5;
4. The communalities were all above 0.3 except for A16, confirming that each A_k except $k=16$ shared some common variance with other items.

The second step was to decide the number of factors and rotation method. Initial eigen values were examined indicating that the first three factors explained 29%, 15%, and 10% of the variance respectively whilst the fourth and fifth factors had eigen values of just over 1, explaining 7% and 6% of the variance respectively. Also, the solutions for 3, 4 and 5 factors were each examined using varimax and oblimin of the factor loading matrix to interpret any correlation between the factors.

A three-factor solution, explaining 54% of the variance and using the oblimin rotation, was chosen given: 'leveling off' of eigen values on the Scree Plot after three factors; an inadequate number of primary loadings; difficulty in

interpreting the fourth and fifth factor; and correlations between factors (0.3) i.e. F1, F3 ($r=.39$) and F2, F3 ($r=.312$). A16 ('Personas I use are usually well formed and adequate.') was eliminated since it failed to meet criterion of a minimum primary factor loading of .35 or above. A6 and A1 were retained even though both contributed to two factors. A6 ('Collaborating around personas is difficult') had a factor loading of 0.412 on F2 (resources for creating personas) and -0.366 on F1 (benefits of personas). One explanation for this is that practitioners see collaboration as a needed resource that presents a challenge from the perspective of persona creation. And from the perspective of practitioners' use, collaboration is not a persona benefit because it is difficult to apply. A1 ('Personas I create/use are time consuming') had a factor loading of 0.382 on F2 (relating to creating personas) and 0.38 on F3 (relating to the representation of personas). This was explained given that the question asked about "create/use" and the percentage of practitioners who create personas was 46% versus 54% who use them.

A PAF of the remaining 15 items using oblimin rotations was conducted, with three factors explaining 56% of the variance. Most items had primary loadings above 0.5 except for A7 and A15. A1 and A6 had cross-loadings into two factors (as explained above). The pattern loading matrix is presented in Table II with items presented in descending order of factor loading to indicate strength and direction with respect to factor. The 3 persona factors were named as follows:

- F1: *Personas benefits*, defined by 5 items positively and 1 item (A7) negatively. The descending order of the factor

loading A12, A11, A13, A14, A15, A7 indicated their strength within their factor.

- F2: *Persona resources*, defined by 6 items positively. The descending order of the factor loading A4, A2, A3, A5, A6, A1 indicated their respective within their factor.
- F3: *Persona pitfalls*, defined by 5 items positively. The descending order of the factor loading A10, A8, A9, A7, A1 indicated their respective within their factor.

TABLE III. FACTOR CORRELATION MATRIX

Factor	<i>Persona benefits</i>	<i>Persona resources</i>	<i>Persona pitfalls</i>
Persona benefits	1.000	-.151	-.366
Persona resources	-.151	1.000	.290
Persona pitfalls	-.366	.290	1.000

TABLE IV. COMPOSITE SCORES FOR PERSONA FACTORS

	<i>Persona benefits</i>	<i>Persona resources</i>	<i>Persona pitfalls</i>
Mean Std.	10.9247	11.6937	9.6429
Deviation	2.35362	2.56391	2.02689
Skewness	-1	-0.41	0.071
Kurtosis	1.729	0.861	-0.677

The reliability of the solution was tested by checking internal consistency of items in each factor. Cronbach's alpha of the 3 factors was good: personas benefits (6 items, $\alpha=.788$), persona resources (6 items, $\alpha=.765$), and persona pitfalls (5 items, $\alpha=.743$). For the first factor, A6 was recoded to remove negative correlation. No substantial increases in alpha for factors could be achieved by eliminating more items. The fourth step computed the composite scores (Table IV) for each factor based on weight sum score of the items and loadings on each factor according to the following equation:

$$\sum_{it=1}^n (FL_{it} * S_{it})$$

where it = items in each factor, FL = Factor Loading, S = Score.

The results of the correlation matrix in TABLE III, along the composite scores in Table IV showed that UX/ID practitioners feel more strongly about the following factors.

1. *Persona resources* was the highest factor, had a left-skewed distribution (Table IV), and was positively correlated ($r \approx 0.3$) with *Persona pitfalls* (Table III). This indicates that practitioners tend to think that persona

resources are ranked first, and an increase in negative attitude towards *persona pitfalls* is likely to occur with an increase in negative attitude towards of *Persona resources*.

2. *Personas benefits* was the second highest factor, had a left-skewed distribution (Table IV) and was negatively correlated ($r=-0.39$) with *persona pitfalls* (Table III). This indicates that practitioners tend to think that benefits of personas come a close second, and an increase in negative attitude towards persona pitfalls is likely to occur with the decrease in positive attitudes to *persona benefits*.
3. *Persona pitfalls* was the third with normal distribution (Table IV), indicating that practitioners tend to think last about the pitfalls which tends to correlate with the previous two factors.

2) Attitudes towards Personas

We report responses to the items in the persona survey, including results of a one-sample sign test used to assess agreement with each proposition. In each of the bar charts (Figure 1–16), the left end of the red arrow indicates the lower bound of the 95% confidence interval and the dot indicates the estimated population median (note that in a number of cases these coincide).

A1: Personas are time-consuming to create/use

Figure 1 shows that the attitudes to this item were mostly positive with median and mode of 4. 62% responded on the 'agree' side of neutral (4 or 5) and 25% responded on the 'disagree' side of neutral (1 or 2). A one-tailed sign test was **highly significant** ($p=.0004$ and $p\text{-adjusted}=0.03125$) supporting H1 (agreement). The odds ratio was 2.5.

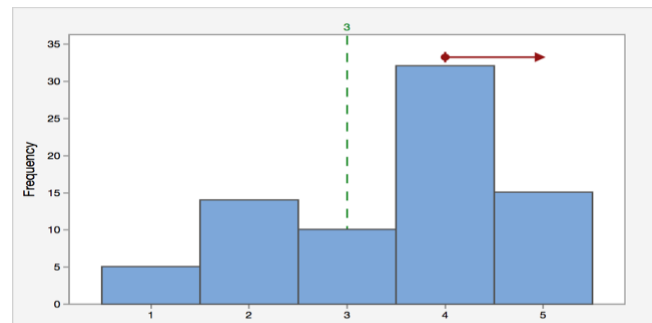


Figure 1. Personas are time-consuming to create/use

Conclusion: Practitioners tend to agree that personas are time-consuming to create/use.

A2: Personas are expensive to create/use

Figure 2 shows that the attitudes to this item were fairly even around neutral with a median of 3 and mode of 4. 34% responded on the 'agree' side of neutral (4 or 5) and 25% responded on the 'disagree' side of neutral (1 or 2). A one-tailed sign test was non-significant (1-tailed $p=.7052$ and $p\text{-adjusted}=0.046875$) supporting H0 (non-agreement). The odds ratio was 0.9.

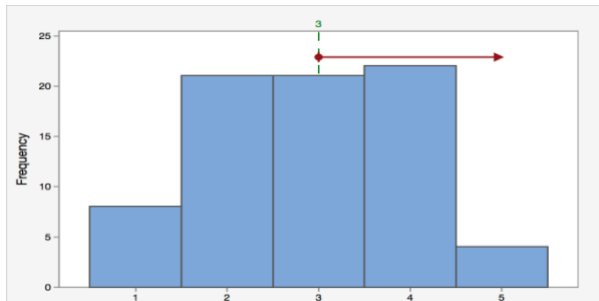


Figure 2. Personas are expensive to create/use

Conclusion: Practitioners tend **not to** agree that personas are expensive to create/use.

A3: Representative personas require a lot of data

Figure 3 shows that the attitudes to this item had a median and mode of 4. 54% responded on the 'agree' side of neutral (4 or 5) and 16% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p < .0001$ and $p\text{-adjusted} = .003125$) supporting H1 (agreement). The odds ratio was 3.4.

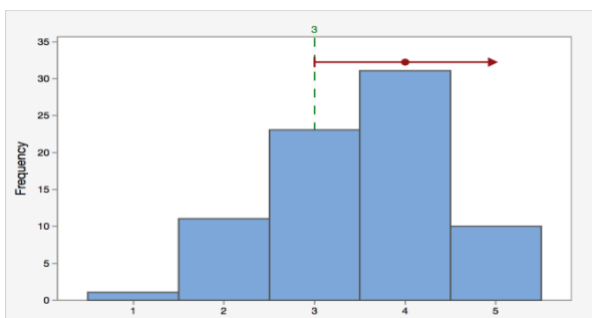


Figure 3. Representative personas require a lot of data

Conclusion: Practitioners tend to agree that representative personas require a lot of data.

A4: Personas require expertise in qualitative research to create.

Figure 4 shows that the attitudes to this item had a median and mode of 4. 72% responded on the 'agree' side of neutral (4 or 5) and 14% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p < .0001$ and $p\text{-adjusted} = .00625$) supporting H1 (agreement). The odds ratio was 5.

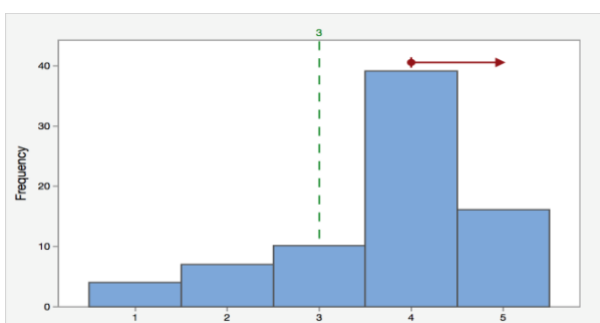


Figure 4. Personas require expertise in qualitative research to create

Conclusion: Practitioners tend to agree that personas require expertise in qualitative research to create.

A5: Personas require training in personas methods

Figure 5 shows that the attitudes to this item had a median and mode of 4. 66% responded on the 'agree' side of neutral (4 or 5) and 13% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant ($Z = 3.846$, 1-tailed $p < .0001$ and $p\text{-adjusted} = .00937$) supporting H1 (agreement). The odds ratio was 5.

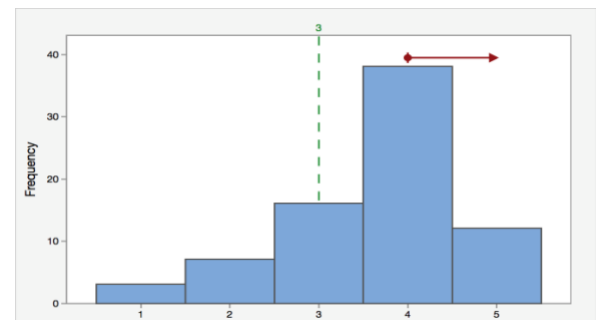


Figure 5. Personas require training in personas methods

Conclusion: Practitioners tend to agree that personas require training in personas methods.

A6: Collaborating around personas is difficult

Figure 6 shows that the attitudes to this item had a median of 3 and mode of 2. 34% responded on the 'agree' side of neutral (i.e., 4 or 5) and 39% responded on the 'disagree' side (1 or 2). A one-tailed sign test was non-significant (1-tailed $p = .748$ and $p\text{-adjusted} = .05$) supporting H0 (neutral or disagree) with an odds ratio ($OR \approx 0.9$).

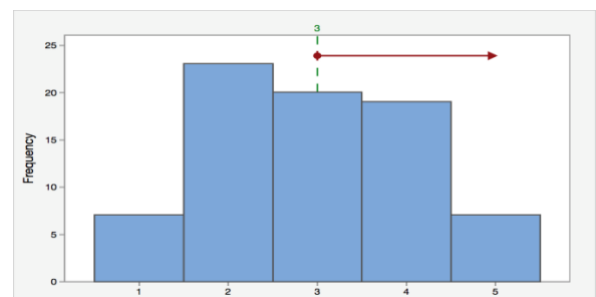


Figure 6. Collaborating around personas is difficult

Conclusion: Practitioners tend not to agree that collaborating around personas is difficult.

A7: Personas are often not properly used by teams

Figure 7 shows that the attitudes to this item had a median of 4 and mode of 5. 78% responded on the 'agree' side of neutral (i.e., 4 or 5) and 4% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p < .0001$ and $p\text{-adjusted} = .00125$) supporting H1 (agreement). The odds ratio was 19.7.

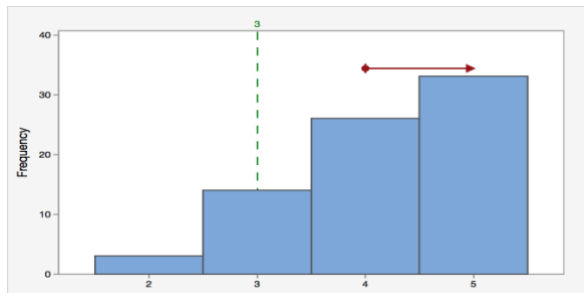


Figure 7. Personas are often not properly used by teams

Conclusion: Practitioners tend to agree that personas are often not properly used by teams.

A8: Personas often represent extreme archetypes

Figure 8 shows that the attitudes to this item had a median and mode of 3. 43% responded on the 'agree' side of neutral (4 or 5) and 22% responded on the 'disagree' side (1 or 2). A one-tailed sign test was found to be significant (1-tailed $p=.025$ and $p\text{-adjusted}=.0344$) supporting H1 (agreement) with an odds ratio of 1.9.

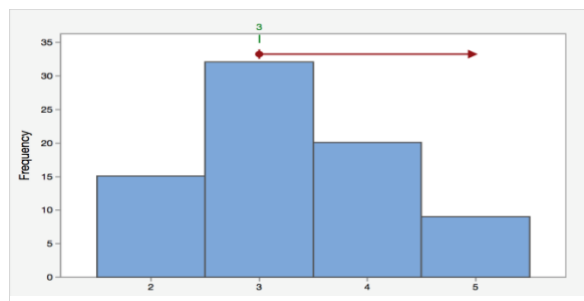


Figure 8. Personas often represent extreme archetypes

Conclusion: Practitioners tend to agree that personas often represent extreme archetypes.

A9: Personas often lack important information related to goals, needs, behaviors, and attitudes.

Figure 9 shows that the attitudes to item had a median and mode of 3. 42% responded on the 'agree' side of neutral (4 or 5) and 26% responded on the 'disagree' side (1 or 2). A one-tailed sign test was found to be non-significant (1-tailed $p=.064$ and $p\text{-adjusted}=.0438$) supporting H0 (neutral or disagree) with an odds ratio of 1.6.

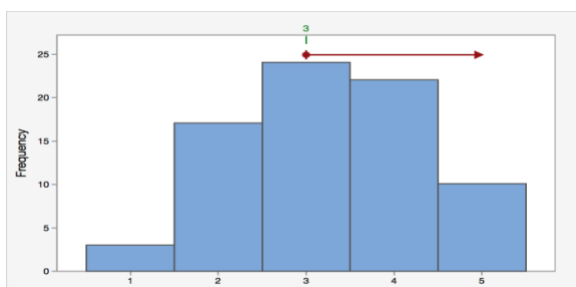


Figure 9. Personas often lack important information related to goals, needs, behaviors, and attitudes.

Conclusion: Practitioners tend not to agree that personas often lack important information related to goals, needs, behaviors, attitudes.

A10: Persona sets often incorporate redundancy

Figure 10 shows that the attitudes to this item had a median and mode of 3. 42% responded on the 'agree' side of neutral (4 or 5) and 26% responded on the 'disagree' side (1 or 2). A one-tailed sign test was found to be significant (1-tailed $p=.064$ and $p\text{-adjusted}=.0438$) supporting H1 (agreement). The odds ratio was 1.8.

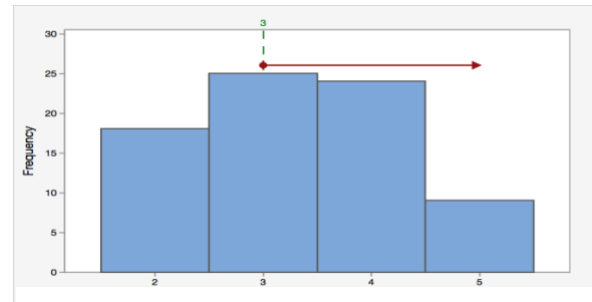


Figure 10. Persona sets often incorporate redundancy

Conclusion: Practitioners tend to agree that personas often incorporate redundancy.

A11: Personas are helpful for understanding users' needs and context

Figure 11 shows that the attitudes to this item had a median and mode of 4. 83% responded on the 'agree' side of neutral (i.e., 4 or 5) and 8% responded on the 'disagree' side. A one-tailed sign test was highly significant (1-tailed $p<.0001$ and $p\text{-adjusted}=.015625$) supporting H1 (agreement). The odds ratio was 10.5.

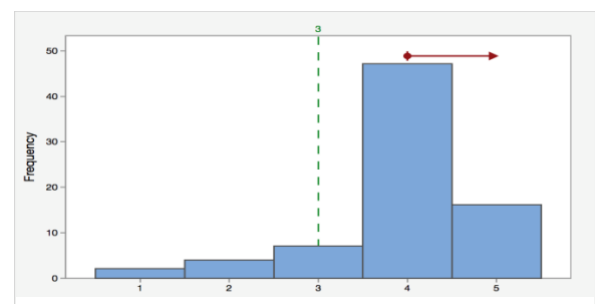


Figure 11. Personas are helpful for understanding users' needs and context

Conclusion: Practitioners tend to agree that personas are helpful for understanding users' needs and context.

A12: Personas are helpful for making design decisions

Figure 12 shows that the attitudes to this item had a median and mode of 4. 72% responded on the 'agree' side of neutral (i.e., 4 or 5) and 11% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p<.0001$).

and $p\text{-adjusted}=.01875$) supporting H1 (agreement). The odds ratio was 6.9.

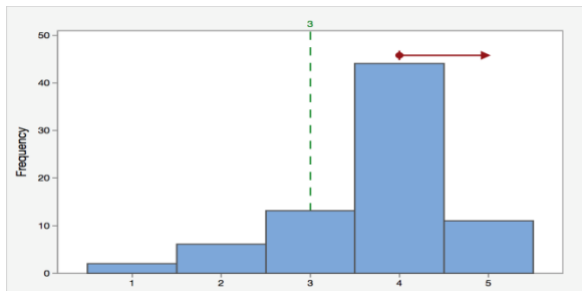


Figure 12. Personas are helpful for making design decisions

Conclusion: Practitioners tend to agree that personas are helpful for making design decisions.

A13: Personas are helpful for implementing and building

Figure 13 shows that the attitudes to this item had a median and mode of 4. 47% responded on the 'agree' side of neutral (4 or 5) and 28% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p=.0318$ and $p\text{-adjusted}=.040625$) supporting H1 (agreement). The odds ratio was 1.7.

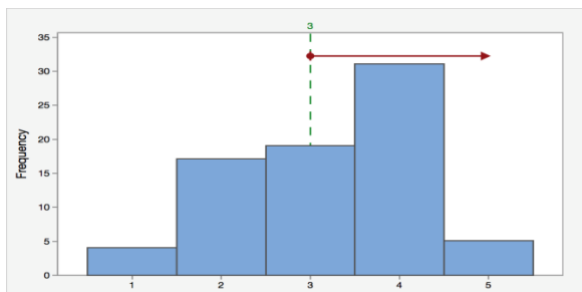


Figure 13. Personas are helpful for implementing and building

Conclusion: Practitioners tend to agree that personas are helpful for implementing and building.

A14: Personas are helpful for evaluation

Figure 14 shows that the attitudes to this item had a median and mode of 4. 68% responded on the 'agree' side of neutral (i.e., 4 or 5) and 12% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p=.0318$ and $p\text{-adjusted}=.021875$) supporting H1 (agreement). The odds ratio was 5.8.

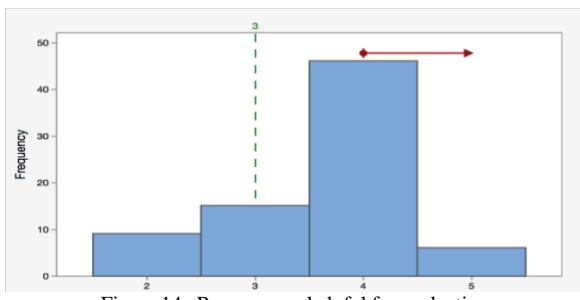


Figure 14. Personas are helpful for evaluation

Conclusion: Practitioners tend to agree that personas are helpful for evaluation.

A15: Personas are helpful for communicating with stakeholders and team members

Figure 15 shows that the attitudes to this item had a median and mode of 4. 75% responded on the 'agree' side of neutral (4 or 5) whilst 11% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p<0.001$ and $p\text{-adjusted}=.025$) supporting H1 (agreement). The odds ratio was 7.1.

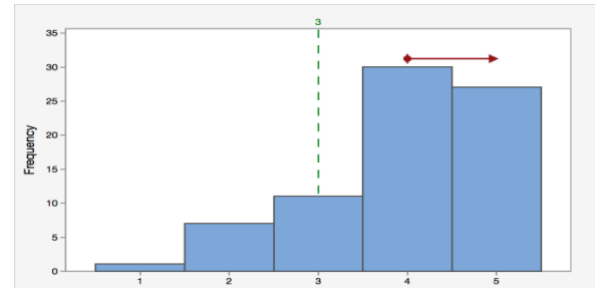


Figure 15. Personas are helpful for communicating with stakeholders and team members

Conclusion: Practitioners tend to agree that personas are helpful for communicating with stakeholders and team members.

A16: Personas I use are usually well formed and adequate

Figure 16 shows that the attitudes to this item had a median and mode of 3. 49% responded on the 'agree' side of neutral (4 or 5) and 16% responded on the 'disagree' side (1 or 2). A one-tailed sign test was highly significant (1-tailed $p=0.003$ and $p\text{-adjusted}=.028125$) supporting H1 (agreement). The odds ratio was 3.

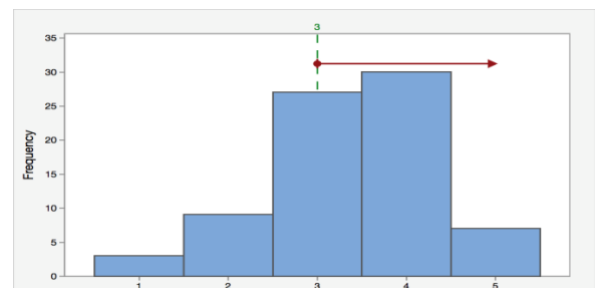


Figure 16. Personas I use are usually well formed and adequate.

Conclusion: Practitioners tend to agree that the personas they use are usually well formed and adequate.

We use the odds ratio to judge relative strength of opinion. Table V shows the items ordered by odds ratio. We use descending order (most strongly held view at the top). The 13 significant items are displayed first followed by the 3 non-significant items (A9, A2, A6).

TABLE V. PRIORITY OF ATTITUDES TOWARDS PERSONAS BASED ON THE DESCENDING ORDER OF OR RATIOS.

Priority	Attitude	OR ratio
1	A7: Personas are often not properly used by teams.	19.6
2	A11: Personas are helpful for understanding users' needs and context	10.5
3	A15: Personas are helpful for communicating with stakeholders and team members	
4	A12: Personas are helpful for making design decisions	7.1
5	A14: Personas are helpful for evaluation	6.9
6	A5: Personas require training in persona methods.	5.8
7	A4: Personas require expertise in qualitative research to create.	5
8	A3: Representative personas require a lot of data.	5
9	A16: The personas I use are usually well formed and adequate.	3.4
10	A1: Personas are time consuming to create/use.	3.1
11	A8: Personas often represent extreme archetypes	2.5
12	A10: Persona sets often incorporate redundancy (multiple personas referring to the same characteristics)	1.9
13	A13: Personas are helpful for implementing and building	1.8
14	A9: Personas often lack important information related to goals, needs, behaviors, and attitudes	1.7
15	A2: Personas are expensive to create/use.	1.6
16	A6: Collaborating around personas is difficult.	0.9

V. CONCLUSION

Existing studies on personas are typically qualitative/ethnographic or case studies. They tend to involve small samples of practitioners with findings developed inductively. These studies are valuable for raising issues, but generalization can be difficult. Also, the claims in the literature are diffused, uncorroborated, and cannot be prioritized. The study reported here addresses these issues by providing a quantitative analysis of the views of a large number of practitioners.

The results show that persona use is quite prevalent amongst IxD/UX practitioners, particularly to capture the results of user research, but also to support design activities and to some extent, to support evaluation. We group attitudes into 3 dimensions (in order of importance): persona resources,

persona benefits, and persona pitfalls. There was a weak negative correlation between persona resources and persona benefits, an acceptable negative correlation between persona benefits and persona pitfalls and an acceptable positive correlation between persona resources and persona pitfalls.

The survey showed that IxD/UX practitioners saw six kinds of resources as being consumed by personas with their order of importance indicated in Table VI. These findings provide a foundation for issues that might have the most impact when considering things like training needs, the design of persona creation methods and the design of persona support tools. Financial costs have also been considered a significant resource costs in the literature, and yet practitioners had an overall neutral opinion towards it. This might be explained by the fact that practitioners are more directly affected by time implications than they are by decisions about budgets.

In terms of persona benefits, practitioners tend to perceive six items with their order of importance indicated in Table VI and collaboration (although not significant for reasons explained earlier) affecting benefits negatively. Our findings on benefits of personas are similar to others and subset of the findings in [8]. Yet, our three highest ranked benefits did not show a difference in opinions between creators and users of personas as suggested by [12][14][18]. These findings indicate that personas are perceived as beneficial for practitioners (creators and users) despite resource concerns and this could provide implications for the priority of benefits that we need our future approaches to focus on.

Practitioners often had strong opinions about challenges that they face with personas. They tend to perceive five kinds of persona pitfalls. The literature has briefly addressed and introduced these (Table VI), but we wanted to explore if these pitfalls are common among practitioners. Our results show that there is one remarkably high ranked pitfall, which is that personas are often not properly used by teams, followed by time required to create them (also found under resources) and other relatively low ranked attitudes in terms of importance. As noted, time was also found under resources and this could be due to ambiguity in the stated term "time consuming to create/ use", which may have been perceived as a resource for the creator, but a pitfall among persona users. This major finding indicates a need for design team members to work together around personas and for this to be addressed in methods and tools. Although collaboration was not explicitly addressed in literature, it was evident in some of the suggested persona development methods and practitioners had an overall neutral opinion towards it. This might be explained, not by a lack of collaboration difficulties, but by a lack of collaboration.

In future work, we plan to follow up on the findings reported here by exploring them more deeply in an interview study with IxD/UX practitioners and to use the findings to support the identification of requirements for persona creation tools. And as educators, given that personas are usually perceived as beneficial in the UCD, we would do well to continue to include personas in our university syllabi but to find approaches that overcome or at least educate students about the challenges of resources and common pitfalls.

TABLE VI. . SUMMARY OF OUR FINDINGS IN COMPARISON WITH LITERATURE (SUPPORTIVE/UN-SUPPORTIVE INDICATES OPINION ELICITED FROM OR OBSERVATION MADE OF REPORTED WORK).

Priority	Factors	Attitude	OR	Supportive	Unsupportive
6	Persona resources (1)	Personas require training in persona methods.	5	[3]–[5][9]–[11][16][20]	
7		Personas require expertise in qualitative research to create.	5	[3][4][10][11][14][18][20]	[7] [8]
8		Representative personas require a lot of data.	3.4	[4][9]–[11][20]	[18][27]
10		Personas are time consuming to create/use.	2.5	[3][8]–[10][12][19][20]	[7][8]
15		Personas are expensive to create/use.	0.9	[4][9]–[11][20][21]	[7][8]
16		Collaborating around personas is difficult	0.9	[4][10][14]	[15]
2	Persona benefits (2)	Personas are helpful for understanding users' needs and context.	10.5	[2]–[7][10][11][20][22]	[14]
3		Personas are helpful for communicating with stakeholders and team members.	7.1	[2]–[6][8][10][11][20][22]	[12]–[14]
4		Personas are helpful for making design decisions.	6.9	[3][4][6][8][10][11][22]	[14]–[16]
5		Personas are helpful for evaluation.	5.8	[11][19][20][22]	[21]
13		Personas are helpful for implementing and building.	1.7	[3][4][6][10][11][20][22]	
16		Collaborating around personas is difficult	0.9	[4][10][14]	[15]
1	Persona pitfalls (3)	Personas are often not properly used by teams.	19.6	[12][13]	[4][8][14]
10		Personas are time consuming to create/use.	2.5	[4][9]–[11][13][20][21]	[7][8]
11		Personas often represent extreme archetypes	1.9	[3][11][22]	
12		Persona sets often incorporate redundancy	1.8	[3] [4]	
14		Personas often lack important information related to goals, needs, behaviors, and attitudes.	1.6	[20]	[2]–[6][10][11]
9		My personas are usually well formed and adequate.	3.1		[21]

REFERENCES

- [1] E. Bashir and S. Attfield, "What's Hot in Interaction Design? An International Survey of Practitioners' Views on Personas," in IARIA, ACHI, 2018, pp. 66-74.
- [2] A. Cooper, *The inmates are running the asylum*. Sams, 1999.
- [3] A. Cooper, R. Reimann, D. Cronin, and A. Cooper, *About face 3 : the essentials of interaction design*. Wiley Pub, 2007.
- [4] J. Pruitt and T. Adlin, *The persona lifecycle : keeping people in mind throughout product design*. Elsevier, 2006.
- [5] L. Nielsen, "Personas," in *The Encyclopedia of Human-Computer Interaction*, 2nd ed., Aarhus: The Interaction Design Foundation, 2014.
- [6] L. Nielsen, K. S. Nielsen, J. Stage, and J. Billestrup, "Going Global with Personas," 2013, pp. 350-357.
- [7] J. Brickey, S. Walczak, and T. Burgess, "Comparing Semi-Automated Clustering Methods for Persona Development," *IEEE Trans. Softw. Eng.*, vol. 38, no. 3, pp. 537-546, May 2012.
- [8] T. Miaskiewicz, T. Sumner, and K. A. Kozar, "A latent semantic analysis methodology for the identification and creation of personas," in *Proceedings of the 26th SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1501-1510.
- [9] J. McGinn and N. Kotamraju, "Data-Driven Persona Development," in *SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 1521-1524.
- [10] J. Pruitt and J. Grudin, "Personas: practice and theory," in *Proceedings of the 2003 conference on Designing for user experiences - DUX '03*, 2003, p. 1.
- [11] S. Mulder and Z. Yaar, *The user is always right : a practical guide to creating and using personas for the Web*. New Riders, 2007.
- [12] E. Friess, "Personas and Decision Making in the Design Process: An Ethnographic Case Study," in *CHI*, 2012.
- [13] T. W. Howard, "Are personas really usable?," *Commun. Des. Q. Rev.*, vol. 3, no. 2, pp. 20-26, Mar. 2015.
- [14] Å. Blomquist and M. Arvola, "Personas in Action: Ethnography in an Interaction Design Team," 2002.
- [15] C. N. Chapman and R. P. Milham, "The Personas' New Clothes: Methodological and Practical Arguments against a Popular Method," *Proc. Hum. Factors Ergon. Soc. Annu. Meet.*, vol. 50, no. 5, pp. 634-636, Oct. 2006.
- [16] T. Matthews, T. Judge, and S. Whittaker, "How do designers and user experience professionals actually perceive and use personas?," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 2012, p. 1219.
- [17] P. Turner and S. Turner, "Is stereotyping inevitable when designing with personas?," *Des. Stud.*, vol. 32, no. 1, pp. 30-44, 2011.
- [18] Y. Chang, Y. Lim, and E. Stolterman, "Personas: From Theory to Practices," in *Proceedings of the 5th Nordic conference on Human-computer interaction building bridges - NordiCHI '08*, 2008, p. 439.
- [19] T. Miaskiewicz and K. A. Kozar, "Personas and user-centered design: How can personas benefit product design processes?," *Des. Stud.*, vol. 32, no. 5, pp. 417-430, 2011.
- [20] C. LeRouge, J. Ma, S. Sneha, and K. Tolle, "User profiles and personas in the design and development of consumer health technologies," *Int. J. Med. Inform.*, vol. 82, no. 11, pp. e251-e268, 2013.
- [21] J. Billestrup, J. Stage, L. Nielsen, and K. S. Hansen, "Persona Usage in Software Development: Advantages and Obstacles," in *The Seventh International Conference on Advances in Computer-Human Interactions Persona*, 2014.
- [22] F. Long, "Real or Imaginary: The effectiveness of using personas in product design | Frontend," in *Proceedings of the IES Conference*, 2009.
- [23] G. Sholmo, "A Closer Look At Personas: A Guide To Developing The Right Ones (Part 2) — Smashing Magazine," *smashingmagazine*, 2014. [Online]. Available: <https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-2/>. [Accessed: 02-Feb-2018].
- [24] G. Sholmo, "A Closer Look At Personas: What They Are And How They Work | 1 — Smashing Magazine," *smashingmagazine*, 2014. [Online]. Available: <https://www.smashingmagazine.com/2014/08/a-closer-look-at-personas-part-1/>. [Accessed: 02-Feb-2018].
- [25] N. Kellingley, "Common Problems with User Personas | Interaction Design Foundation," 2015. [Online]. Available: <https://www.interaction-design.org/literature/article/common-problems-with-user-personas>. [Accessed: 10-Nov-2018].
- [26] P. Bryan, "Are Personas Still Relevant to UX Strategy? :: UXmatters," *UXmatters*, 2013. [Online]. Available: <https://www.uxmatters.com/mt/archives/2013/01/are-personas-still-relevant-to-ux-strategy.php>. [Accessed: 10-Nov-2018].
- [27] D. Norman, "Ad-Hoc Personas & Empathetic Focus - jnd.org," *jnd.org*, 2004. [Online]. Available: http://jnd.org/dn.mss/adhoc_personas_empathetic_focus.html. [Accessed: 10-Nov-2018].
- [28] E. Sonderer, R. Sanderman, and J. C. Coyne, "Ineffectiveness of Reverse Wording of Questionnaire Items: Let's Learn from Cows in the Rain," *PLoS One*, vol. 8, no. 7, p. e68967, Jul. 2013.
- [29] C. A. Schriesheim and K. D. Hill, "Educational and Psychological Measurement Controlling Acquiescence Response Bias by Item Reversals: the Effect on Questionnaire Validity," *Sociol. Methods Res.*, vol. 21, no. 1, pp. 52-88.
- [30] J. A. Ferreira, J. A. Ferreira, and A. H. Zwinderman, "On the Benjamini-Hochberg method," *ANN. Stat.*, pp. 1827-1849, 2006.

Context-aware Storage and Retrieval of Digital Content

Database Model and Schema Considerations for Content Persistence

Hans-Werner Sehring

Namics

Hamburg, Germany

e-mail: hans-werner.sehring@namics.com

Abstract—In increasingly many information systems that publish digital content, the documents that are generated for publication are tailored for and delivered to users working in different and varying contexts. To this end, the content from which an actual document is created is dynamically selected with respect to a specific context. The task of content selection incorporates queries to an underlying database that hosts data representing content. Such queries are parameterized with a description of the context at hand. This is particularly true for content management applications, e.g., for websites that are targeted at a user's context. The notion of context comprises various dimensions of parameters like language, location, time, user, and user's device. Most data modeling languages, including programming languages, are not well prepared to cope with variants of content, though. They are designed to manage universal, consistent, and complete sets of data. The Minimalistic Meta Modeling Language (M3L) can be applied as a language for content representation. M3L has proven particularly useful for modeling content in context. Towards an operational M3L execution environment, we are researching mappings to databases of different data models, and for each data model schemas to efficiently store and utilize M3L models. This article discusses such schemas for context-aware data representation and retrieval. The main focus lies on efficiency of queries used for M3L evaluation with the goal of context-dependent content selection. This is achieved by expressing context-aware models, in particular M3L statements, by means of existing persistence technology.

Keywords—data modeling; data schema; databases; content modeling; context-aware data modeling; content; content management; content management systems; context.

I. INTRODUCTION

In many information systems, e.g., web-based ones, data represents *content* to be incorporated into *documents* that are generated on purpose. More often than not content is required to be queried dynamically on document access, calling for adequate content storage and retrieval. First studies on such content persistence have been reported [1]. This article extends the report on the current state of these database schema investigations.

In the digital society [2], data is required to represent all kinds of content, ranging from structured content of text documents to unstructured, typically binary representations of

video and audio content. Content is used for many purposes, the most obvious ones being information and commerce. Content is published by means of documents, often multimedia documents incorporating different media that are interrelated to form hypermedia networks. So-called publication channels offer the medium for one kind of publication, e.g., a website, a document file, or a mobile app. Content is typically represented in a channel-agnostic way in order to support multi- or even omni-channel publishing.

It is quite common to deliver content to users in a way that addresses the *context* in which they are when requesting the content. This may include the channel they are using, the working mode they are in, the history of previous usage scenarios, etc. Targeting content to users' contexts can range from simply arranging content in a specific way, over specifically assembled documents, to content that is synthesized for the current requests. Examples are a prominent display of teasers for content that is assumed to be of interest to the user, the production of documents matching a user's native language, adjustment of document quality based on the current network bandwidth and the receiving device, and creating content that represents some base data in knowledgeable form.

For such content targeting scenarios, data needs to be stored in a way that allows generating different views on the content, mainly by selecting content relevant in a certain context. Data representing all forms of content in such a system, therefore, needs to be attributed with the contexts in which it is applicable or preferred. Obviously, some notion of context is required for such representations [3].

Data modeling and programming languages typically do not exhibit features to represent context and to include it in evaluations. Database management systems, being the backbone of practically every information system, are particularly optimized for one connected set of data that is supposed to be consistent and complete. This means that they are not well equipped for dynamic content production, neither regarding content representation nor efficient context-dependent retrieval.

Data retrieval needs particular attention when content is dynamically assembled depending on some context in which it is requested. For the tasks of context-aware content management, complex collections of data to be used as content are requested frequently. A context-aware schema has

to efficiently support the underlying queries that are employed to identify relevant content.

For the discussion of data models, we consider content in contexts as it is expressible using the Minimalistic Meta Modeling Language (M3L). This language allows expressing content in a straightforward way. Being a modeling language, there is no obvious mapping to established data structures, though.

The rest of this paper is organized as follows. Section II reviews related work in the area of context-aware data and content models. Sections III and IV give a brief overview over the M3L and describe those parts of the language that are required for the discussion in this paper. Section III describes the static aspects of the M3L used to define application models. Section IV focuses on the dynamic evaluation of such models. The architecture of a current M3L implementation is discussed in Section V in order to clarify the scope of M3L persistence. Section VI presents a first conceptual model of an internal representation of M3L concepts. Section VII makes this model more concrete by means of logical representations, comparable to the logical view on databases. Aspects of M3L persistence implementations based on different data models are touched in Section VIII. The conclusion and acknowledgment close the paper in Section IX.

II. RELATED WORK

Context is important in the area of content management, but also in other modeling domains. This section names some existing modeling approaches to contextual information.

A. Content Management Products

Most commercial content management products have introduced some notion of context in their models and processes. They utilize context information to *target* content to users. Some use the term *personalization*, which is similar to, but different from contextualization [4].

In most cases, there are publication *rules* associated with content, similar as discussed in [5]. These rules are based on so-called *segments*. Every user is assigned one or more segments. When requesting content, the rules are evaluated for the actual segment(s) in order to select suitable content.

Content authors and editors maintain the content rules. Segments are assigned to users automatically by the systems based on the users' behavior (user interactions), the user journey (e.g., previously visited sites and search terms used for finding the current website), and context information (e.g., device used and location of the user).

Segments offer a rather universal notion of context, though there is no explicit context model.

B. Context-aware Data Models

Parallel to the notion of context used for content, there exists some work on the influence of environments on running applications. In mobile usage scenarios, context refers mainly to such environmental considerations, e.g., network availability, network bandwidth, device, or location.

Context changes are incorporated dynamically into evaluations in these scenarios [6].

Context-awareness is not limited to data models. It is also used for adaptable or adaptive software systems, e.g., to map software configurations to execution environments [7], or to control the behavior of a generic solution [8].

C. Concept-oriented Content Management

Concept-oriented Content Management (CCM) [9] is an approach to manage content reflecting knowledge. Such content does not represent simple facts, but instead is subject to interpretation. Furthermore, the history of things is described by content, not just their latest state.

CCM is not directly concerned about modeling context. Instead, it aims to introduce a form of pragmatics into content modeling that allows users on the one hand to express differing views by means of individual content models, and on the other hand to still communicate by exchanging content between individualized models.

CCM uses a notion of personalization that goes far beyond the one of content management systems (see above).

It is similar to contextualized content usage, although the system does not know about the context of a user. Instead, users carry out personalization (in CCM terms) manually.

A CCM system reacts to model changes and relates model variants to each other. The basis for this ability is systems generation: based on the definitions of users, schemas, APIs, and software modules are generated.

Some aspects of the considerations presented in Section VIII were gained from the research on the generation of CCM modules that map content to external data, e.g., content representations stored in databases.

III. THE MINIMALISTIC META MODELING LANGUAGE

The *Minimalistic Meta Modeling Language* (M3L, pronounced "mel") is a modeling language that is applicable to a range of modeling tasks. It proved particularly useful for context-aware content modeling [10].

For the purpose of this paper, we only introduce the static aspects of the M3L in this section. Dynamic evaluations that are defined by means of different rules are presented in the subsequent section.

The descriptive power of M3L lies in the fact that the formal semantics is rather abstract. There is no fixed domain semantics connected to M3L definitions. There is also no formal distinction between typical conceptual relationships (specialization, instantiation, entity-attribute, aggregation, materialization, contextualization, etc.).

A. Concept Definitions and References

A M3L definition consists of a series of definitions or references. Each definition starts with a previously unused identifier that is introduced by the definition and may end with a semicolon, e.g.:

```
Person;
```

A reference has the same syntax, but it names an identifier that has already been introduced.

We call the entity named by such an identifier a *concept*.

The keyword *is* introduces an optional reference to a *base concept*, making the newly defined concept a *refinement* of it.

A specialization relationship as known from object-oriented modeling is established between the base concept and the newly defined derived concept. This relationship leads to the concepts defined in the context (see below) of the base concept to be visible in the derived concept.

The keyword `is` always has to be followed by either `a`, `an`, or `the`. The keywords `a` and `an` are synonyms for indicating that a classification allows multiple sub-concepts of the base concept:

```
Peter is a Person; John is a Person;
```

There may be more than one base concept. Base concepts can be enumerated in a comma-separated list:

```
PeterTheEmployee is a Person, an Employee;
```

The keyword `the` indicates a closed refinement: there may be only one refinement of the base concept (the currently defined one), e.g.:

```
Peter is the FatherOfJohn;
```

Any further refinement of the base concept(s) leads to the redefinition (“unbinding”) of the existing refinements.

Statements about already existing concepts lead to their redefinition. For example, the following expressions define the concept `Peter` in a way equivalent to the above variant:

```
Peter is a Person;
Peter is an Employee;
```

B. Content and Context Definitions

Concept definitions as introduced in the preceding section are valid in a context. Definitions like the ones seen so far add concepts to the top of a tree of contexts. Curly brackets open a new context, e.g.:

```
Person { Name is a String; }
Peter is a Person{"Peter Smith" is the Name;}
Employee { Salary is a Number; }
Programmer is an Employee;
PeterTheEmployee is a Peter, a Programmer {
  30000 is the Salary;
}
```

We call the outer concepts the *context* of the inner, and we call the set of inner concepts the *content* of the outer.

In this example, we assume that concepts `String` and `Number` are already defined. The sub-concepts created in context are unique specializations in that context only.

As indicated above, concepts from the context of a concept are inherited by refinements. For example, `Peter` inherits the concept `Name` from `Person`.

M3L has visibility rules that correlate to both contexts and refinements. Each context defines a scope in which defined identifiers are valid. Concepts from outer contexts are visible in inner scopes. For example, in the above example the concept `String` is visible in `Person` because it is defined in the topmost scope. `Salary` is visible in `PeterTheEmployee` because it is defined in `Employee` and the context is inherited. `Salary` is not valid in the topmost context and in `Peter`.

C. Contextual Amendments

Concepts can be redefined in contexts. This happens by definitions as those shown above. For example, in the context of `Peter`, the concept `Name` receives a new refinement.

Different aspects of concepts can explicitly be redefined in a context, e.g.:

```
AlternateWorld {
  Peter is a Musician {
    "Peter Miller" is the Name;
  }
}
```

We call a redefinition performed in a context different from that of the original definition a *conceptual amendment*.

In the above example, the contextual variant of `Peter` in the context of `AlternateWorld` is both a `Person` (initial definition) and a `Musician` (additionally defined). The `Name` of the contextual `Peter` has a different refinement.

A redefinition is valid in the context it is defined in, in sub-contexts, and in the context of refinements of the context (since the redefinition is inherited as part of the content).

D. Concept Narrowing

There are three important relationships between concepts in M3L.

M3L concept definitions are passed along two axes: through visibility along the nested contexts, and through inheritance along the refinement relationships.

A third form of concept relationship, called *narrowing*, is established by dynamic analysis rather than by static definitions like content and refinement.

For a concept c_1 to be a narrowing of a concept c_2 , c_1 and c_2 need to have a common ancestor, and they have to have equal content. Equality in this case means that for each content concept of c_2 there needs to be a concept in c_1 's content that has an equal name and the same base classes.

For an example, assume definitions like:

```
Person { Sex; Status; }
MarriedFemalePerson is a Person {
  Female is the Sex;
  Married is the Status;
}
MarriedMalePerson is a Person {
  Male is the Sex;
  Married is the Status;
}
```

With these definitions, a concept

```
Mary is a Person {
  Female is the Sex;
  Married is the Status;
}
```

is a narrowing of `MarriedFemalePerson`, even though it is not a refinement of that concept, and though it introduces separate nested concepts `Female` and `Married`.

E. Semantic Rule Definitions

For each concept, one *semantic rule* may be defined.

The syntax for semantic rule definitions is a double turnstile (“ \models ”) followed by a concept definition. A semantic rule follows the content part of a concept definition, if such exists.

A rule's concept definition is not made effective directly, but is used as a prototype for a concept to be created later.

The following example redefines concepts `MarriedFemalePerson` and `MarriedMalePerson`:

```

MarriedFemalePerson is a Person {
  Female is the Sex;
  Married is the Status;
} |= Wife
MarriedMalePerson is a Person {
  Male is the Sex;
  Married is the Status;
} |= Husband

```

The concepts *wife* and *Husband* are not added directly, but at the time when the parent concept is evaluated. Evaluation is covered by the subsequent section.

Concepts from semantic rules are created and evaluated in different contexts. The concept is instantiated in the same context in which the concept carrying the rule is defined. The context for the evaluation of a rule (evaluation of the newly instantiated concept, that is) is that of the concept for which the rule was defined.

In the example above, the concept *wife* is created in the root context and is then further evaluated in the context of *MarriedFemalePerson*.

Rules are passed from one concept to another by means of inheritance. They are passed to a concept from (1) concepts the concept is a narrowing of, and (2) from base classes. Inheritance happens in this order: Only if the concept is not a narrowing of a concept with a semantic rule then rules are passed from base concepts.

E.g., *Mary* as defined above evaluates to *wife*.

F. Syntactic Rule Definitions

Additionally, for each concept one *syntactic rule* may be defined.

Such a rule, like a grammar definition, can be used in two ways: to produce a textual representation from a concept, or to recognize a concept from a textual representation.

A semantic rule consists of a sequence of string literals, concept references, and the name expressions that evaluate to the current concept's name.

During evaluation of a syntactic rule, rules of referenced concepts are applied recursively. Concepts without a defined syntactic rule are evaluated to/recognized from their name.

E.g., from definitions

```

WordList {
  Word; Remainder is a WordList;
} |- Word " " Remainder;
OneWordWordList is a WordList |- Word;
Sentence { WordList; } |- WordList "."
HelloWorld is a Sentence {
  Words is the WordList {
    Hello is the Word;
    OneWordWordList is the Remainder {
      World is the Word;
    } } }

```

the textual representation

Hello World.

is produced.

Syntactic rule evaluation is not covered in this article.

IV. CONCEPT EVALUATION

As pointed out, there is no fixed generic semantic of M3L constructs. Nevertheless, concrete models receive semantics

by means of semantic rules and their evaluation. After definition, each concept (in the root context) is evaluated in a way described in this section.

Concept evaluation is based on (a) narrowing (see Section III.D) and (b) semantic rules (Section III.E).

This section gives a semi-formal description of these means to assign semantics to M3L models. We present as many definitions as are required to derive the main database operations that drive the evaluation process in database-driven M3L implementations.

Throughout this section, let \mathbb{C} be the set of concepts, \mathbb{S} be the set of sets of concepts, and \mathbb{R} be the set of semantic rules. Let \mathbb{T} be the set of root concepts (concepts that do not have another concept as their explicit context).

A. Concept Relationship Access Functions

First, we define typical access functions to the components of a M3L model.

The function *context* returns the context of a concept as defined by a concept definition, or \perp , if the given concept is a root concept:

$$\text{context}: \mathbb{C} \rightarrow \mathbb{C}. \quad (1)$$

The reverse relation, *content*, returns the content of a concept:

$$\begin{aligned} \text{content}: \mathbb{C} \rightarrow \mathbb{S}: c \mapsto \{c' \in \mathbb{C} \mid \text{context}(c') = c\}, c \neq \perp, \\ \text{content}: \mathbb{C} \rightarrow \mathbb{S}: \perp \mapsto \mathbb{T}. \end{aligned} \quad (2)$$

The *base* relationship maps a concept to its base concepts:

$$\text{base}: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{S}. \quad (3)$$

Since the set of base concepts may be extended by contextual concept amendments, the relation is evaluated relative to a context, given by the context-defining concept (second parameter), or by \perp if base concepts as defined in the root context are requested.

The inverse, the *refine* relationship, maps concepts to the concepts derived from them in a given context x :

$$\text{refine}: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{C}: (c, x) \mapsto \{c' \in \mathbb{C} \mid c' \in \text{base}(c, x)\}. \quad (4)$$

Let *semanticRule* be a projection function that returns the semantic rule defined for a concept in a given context x . If none is defined in x or any parent context, the function returns \perp .

$$\text{semanticRule}: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{R} \quad (5)$$

Likewise, let *concept* be the function that returns the concept that is defined by a rule definition:

$$\text{concept}: \mathbb{R} \rightarrow \mathbb{C}. \quad (6)$$

E.g., for a concept *Concept* in the root context defined as

Concept |= NewConcept {Content;}

the function application $concept(semanticRule(Concept, \perp))$ returns $NewConcept$.

B. Computed Relationships

On the basis of the accessor functions defined in the previous subsection, some computed relationships can be defined. In this subsection we define helper functions required to define narrowing in the subsequent subsection, and to finally define evaluation in Section IV.D: the set of transitive base concepts $base^T$, the set of transitive refinements $refine^T$, and the *bottom* of a concept set.

Chained base relationships are retrieved by

$$base^T: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{S},$$

$$base^T: (c, x) \mapsto base(c, x) \cup \{base^T(c', x) \mid c' \in base(c, x)\}. \quad (7)$$

Likewise, transitive refinement is defined by:

$$refine^T: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{S},$$

$$refine^T: (c, x) \mapsto \{c\} \cup \{refine^T(c', x) \mid c' \in refine(c, x)\}. \quad (8)$$

The function *bottom* removes concepts from a concept set if these are already subsumed by other contained concepts. These are concepts that are refined by a concept in the set and are themselves not refining that concept:

$$bottom: \mathbb{S} \times \mathbb{C} \rightarrow \mathbb{S},$$

$$bottom: (S, x) \mapsto S \setminus \{c \in S \mid \exists c_2 \in S: c_2 \in refine^T(c, x) \wedge c \notin refine^T(c_2, x)\}. \quad (9)$$

C. Concept Narrowing

One central point in the process of evaluating concepts is to compute their narrowing. In order to define narrowings, we first introduce some helper functions.

Let R_c be the set of root concepts that (transitively) are base concepts of a concept c , $R_c = \mathbb{T} \cap base^T(c, x)$. A superset of c 's narrowing is easily computed using

$$narrowCandidateList: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{S}$$

$$narrowCandidateList: (c, x) \mapsto \{refine^T(c', x) \mid c' \in R_c\}, \quad (10)$$

meaning that all narrowings are found in the set consisting of all concepts from all content hierarchies to which the concept belongs.

In order to remove candidates for narrowings, helper functions to examine a concept's "type" are required. Two functions help analyzing whether a concept c is a refinement of a base concept b , (interpreted in the context of concept x):

$$hasType: \mathbb{C} \times \mathbb{C} \times \mathbb{C} \rightarrow \text{Bool}$$

$$hasType: (c, b, x) \mapsto base^T(c, x) \supseteq base^T(b, x), \quad (11)$$

and whether two concepts c_1 and c_2 are the same with respect to their set of base concepts:

$$sameType: \mathbb{C} \times \mathbb{C} \times \mathbb{C} \rightarrow \text{Bool}: (c_1, c_2, x) \mapsto c_2 \in base^T(c_1, x) \vee c_1 = c_2 \vee hasType(c_1, c_2, x). \quad (12)$$

Besides these static type checks, we also need structural matching of concepts (sometimes called "duck typing" [11]):

$$hasWholeContent: \mathbb{C} \times \mathbb{C} \times \mathbb{C} \rightarrow \text{Bool}$$

$$hasWholeContent: (c, candidateBaseConcept, x) \mapsto \forall c_1 \in content(candidateBaseConcept): \exists c_2 \in content(c): sameType(c_2, c_1, x). \quad (13)$$

The function *hasWholeContent* determines for two concepts c and *candidateBaseConcept* whether (interpreted w.r.t. the context of concept x) the whole content of c is also part of the context of *candidateBaseConcept*, meaning that there is a concept with an equal set of base classes.

With the helper functions (10)–(13) we define the narrowing of a concept c in the context of a concept x as:

$$narrowing: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{S}: (c, x) \mapsto refine^T(c, x) \cup \{c' \in narrowCandidateList(c, x) \mid hasType(c, c', x) \wedge hasWholeContent(c, c', x)\}. \quad (14)$$

D. Semantic Rule Application and Concept Evaluation

At the core of the concept evaluation lies the productive application of semantic rules as described in Section III.E.

During the evaluation process, semantic rules are applied by instantiating the concept named in a rule. We express this by a function *apply* as

$$apply: \mathbb{R} \times \mathbb{C} \rightarrow \mathbb{C}:$$

$$apply: (r, x) \mapsto concept(r) \text{ in } context(x), \text{ if it exists,}$$

$$apply: (r, x) \mapsto \text{deep copy of } concept(r) \text{ in } context(x), \text{ interpreted in } x, \text{ else.} \quad (15)$$

With narrowing and rule application we can define M3L concept evaluation as

$$evaluate: \mathbb{C} \times \mathbb{C} \rightarrow \mathbb{S},$$

$$evaluate: (c, x) \mapsto bottom(evaluate(apply(semanticRule(narrowing(c, x), x), x), x),$$

$$\text{if some concept in } narrowing(c, x) \text{ has a rule,}$$

$$evaluate: (c, x) \mapsto bottom(evaluate(refine^T(c, x), x),$$

$$\text{if some concept in } refine^T(c, x) \text{ has a semantic rule}$$

$$evaluate: (c, x) \mapsto bottom(refine^T(c, x), x), \text{ else.} \quad (16)$$

For the sake of brevity, we use extensions to set-valued parameters to relationships (5), (15), and (16).

V. ANATOMY OF THE M3L ENVIRONMENT

This section outlines the architecture of a first M3L implementation. It is studied here in order to determine base functions that require an efficient implementation for concept evaluation. This leads to the requirements on the persistence layer that is the subject of this article.

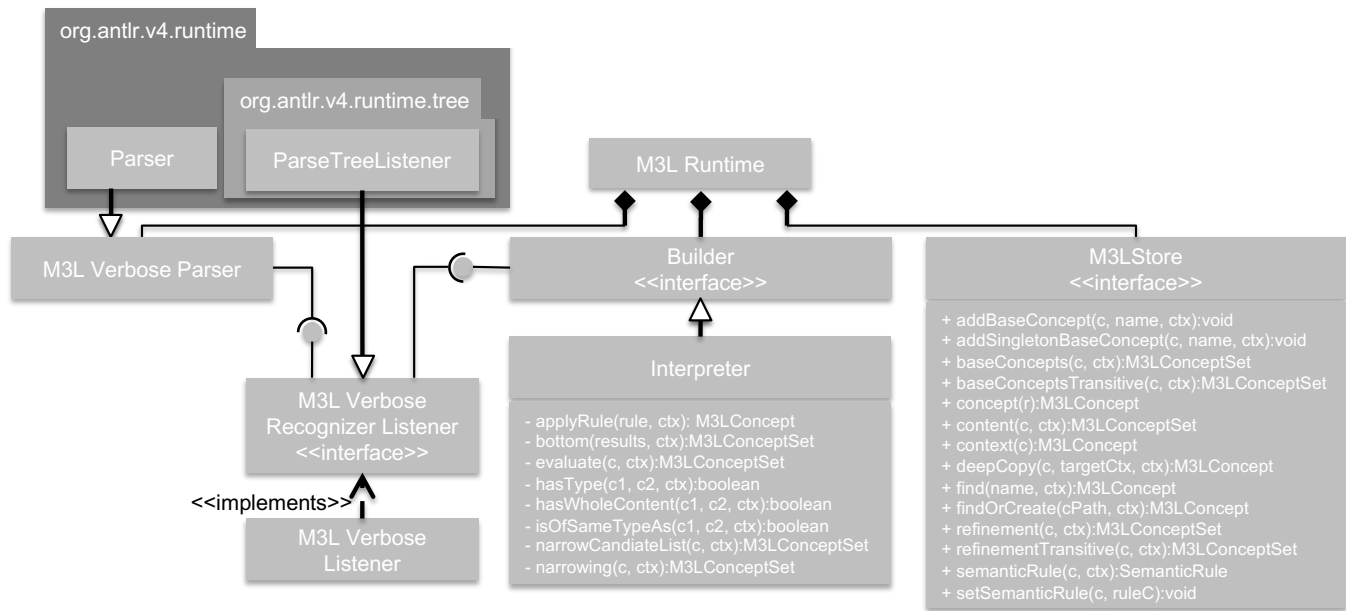


Figure 1. Architecture of the current M3L implementation.

When implementing concept evaluation (16) and all supporting functions (1)-(15), one notices that there are the basic accessor functions (1)-(6). Other functions are defined on top of these basic functions. Therefore, an efficient implementation will place the accessor functions (1)-(6) and those making heaviest use of them, (7) and (8), close to the data layer, while the others can be implemented in a storage-agnostic way. These assumptions lead to an architecture as the one presented in this section.

The current M3L runtime environment is an application that is based on several components. These components are interchangeable in order to be applicable in a wide range of configurations, namely different M3L syntaxes (compact or verbose), interpretation of files or interactive input, or compilation to different target languages and different persistence technology for concept storage and retrieval.

The UML class diagram in Fig. 1 illustrates this M3L implementation. For the method signatures shown in the diagram assume *M3LConcept* to be an interface for concept representations and *M3LConceptSet* to be a set of those.

For brevity, the types of method arguments are omitted. In the figure, *c* denotes a *M3LConcept* to perform an operation on, *ctx* a *M3LConcept* giving the context of the operation, *name* a String giving a concept name, and *r* a semantic rule.

At the frontend, a Parser recognizes M3L statements and creates an abstract syntax tree (AST) for further processing. The parser is based on a parser generated by the *ANTLR* (*ANother Tool for Language Recognition*) parser generator [12]. The grammar of the M3L is quite simple. Still, this powerful parser generator is employed because it plays an important role for the handling of syntactic rules (see Section III.F) at runtime and thus is part of the setup anyway.

In fact, there are different parsers and listeners for different syntaxes of the M3L we are experimenting with. Fig. 1 shows the *M3LVerboseParser* for the syntax used in this article.

In the next stage of M3L processing, a *Builder* creates an internal representation of the parsed M3L definitions.

Using the AntLR framework, a *Parser* and a *Builder* are connected by an observer, here the *M3LVerboseListener*, that receives callbacks whenever the parser recognized a syntactic construct.

In order to receive notifications, the observer implements methods defined by the AntLR API in the interface *ParseTreeListener*. The interfaces are not shown in detail but illustrated in UML by the “lollipop”. In turn an observer uses an interface provided by *Builder* implementations (again represented by a lollipop) to pass information to them.

These interfaces allow different *Builder* implementations. Most notably, there are interpreters and compilers. The *Interpreter* acts directly. It contains generic code for the creation and evaluation of concepts. This code is based on operations provided by a *M3LStore* (see below). The inner working of the *Interpreter* is outlined by the private methods shown on the diagram in Fig. 1. The methods implement those functions from Section. IV that are expressed using the more basic functions.

A compiler creates equivalent code for the creation and evaluation of concepts that can (repeatedly) be executed.

Every concrete *Builder* implements the methods defined in the *Builder* interface that decorate the AST and pass the intermediate representation to a *M3LStore*. These methods are omitted in Fig. 1 in the shown *Interpreter*. Additionally, concrete builder implementations typically define methods for the functions (9)-(16) for concept evaluation. In Fig. 1 such methods are listed as private methods of *Interpreter*.

Analysis of these functions unveils the functionality to be provided by a *M3LStore*. According to this design, *M3LStore* implementations deliver the base functionality required for the builders, namely the required access functions as well as computed relationships that use them most (1)-(8).

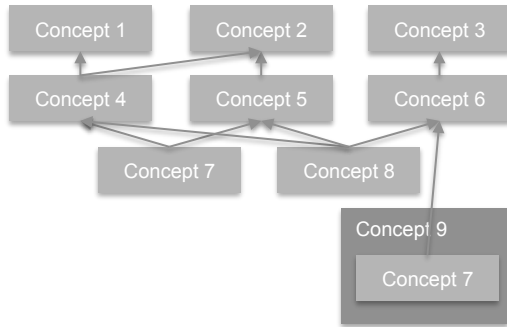


Figure 2. M3L concept refinements and contexts.

The *M3LStore* interface shown in Fig. 1 consists of methods used by a *Builder* to build up a model during parsing, and the abovementioned methods that implement the base functions used during concept evaluation (1)-(8).

For an efficient implementation, we lay an emphasis on the responsibilities of the *M3LStore*. The remainder of this article discusses mappings to some established persistence technologies that can be used as a basis of *M3LStore* implementations.

VI. A CONCEPTUAL MODEL FOR CONTENT REPRESENTATIONS

A conceptual model, as known from database modeling, serves as a first step towards data models for context-aware content. The notion of “concept” is ambiguous here: The aim is a model of (M3L) concepts. A conceptual model for this allows us to abstract from the M3L as a language. The model is not supposed to address practical properties such as operational complexity.

A set of M3L concept definitions can be viewed as a graph with each node representing a concept, labeled with the name of the concept. There are two kinds of edges to represent specialization and contextualization. In fact, such a graph forms a hypergraph to account for contextualization. Every node can contain a graph reflecting definitions as the concept’s content.

The following subsections detail specialization and contextualization relationships, as well as contextual redefinitions.

A. Representing Specialization

Conceptually, a specialization/generalization relationship can straightforward be seen as a many-to-many relationship between concepts. Fig. 2 shows an example.

Arrows with filled heads, directed from a concept to its base concepts, represent specialization relationships in the figure. For example, *Concept 4* is a refinement of *Concept 1* and *Concept 2*.

Fig. 2 furthermore indicates an amendment in a context, namely *Concept 9*. While *Concept 7* is a refinement of *Concept 4* and *Concept 5* in the default context, it is additionally a refinement of *Concept 6* in the context of *Concept 9* (if it is an *is a*/is an definition; otherwise, *Concept 7* would only be a refinement of *Concept 6* in the context of *Concept 9*).

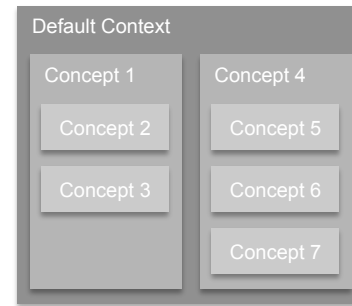


Figure 3. M3L concept definitions in contexts.

B. Representing Context

Since contexts form a hierarchy, contextualization can be represented by a one-to-many relationship between concepts in the roles of context and content.

Fig. 3 represents such a hierarchy by nested boxes shown for concepts. The contextualization relationship is thus visually represented by containment. For example, *Concept 2* is part of the content of *Concept 1*, or *Concept 2* is defined in the context of *Concept 1*.

The outermost context is the default context. There is no corresponding concept for this context.

C. Representing Contextual Information

Specialization and contextualization act together. Refinements of a concept inherit its content; concepts from that content are valid also in the context of the refinement. Each context allows concept amendments. These are a second way to add variations of concepts.

In order to represent contextualized redefinitions, we introduce two kinds of context definitions: *Initial Concept Definition* and *Contextual Concept Amendment*. Both can be placed in any context.

An initial concept definition is placed in the topmost context in which a concept is defined. Redefinitions of concepts are represented by contextual amendments inside the concept in whose context the redefinition is performed.

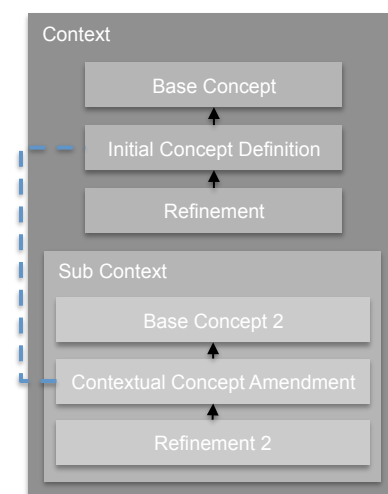


Figure 4. M3L concept amendments in contexts.

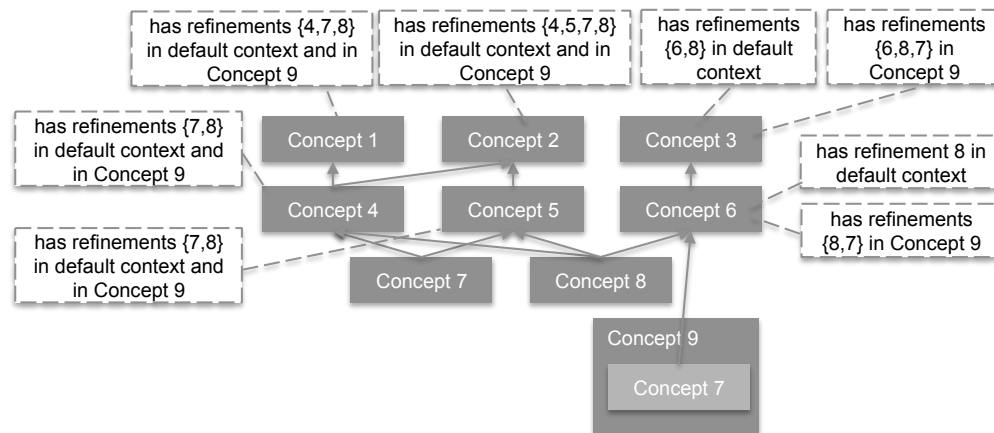


Figure 5. Representation of refinements using materialized transitive refinement relationships.

Fig. 4 illustrates such a concept redefinition scenario. As before, contexts are depicted as nested boxes. There is one *Context* and a *Sub-Context*. Both show a *Concept* that has originally been defined as a refinement of *Base Concept* and is itself refined to *Refinement*. In the context on *Sub-Context*, the concept gets the additional base concept *Base Concept 2*, and there is another refinement *Refinement 2*. These additions are recorded in the contextual amendment of *Concept* in *Sub-Context*. This is, of course, transparent on model level.

Amendments have a reference to the next higher definition. This reference is called *Original*. In Fig. 4, it is shown by the dotted line.

Traversal of the original references allows collecting all definitions in order to determine the effective definition.

VII. LOGICAL CONTENT REPRESENTATION

This section refines contextual content representation models to a level similar to that of a logical data model. This way it discusses properties of data representations without taking implementation details into account.

The complexity of lookups is of major importance for the schema design. During the evaluation of M3L statements, many graph traversals are required to find all valid contexts, all base concepts (to determine content sets) and all refinements (to narrow down concepts before applying rules; this evaluation process is not laid out in this paper).

The most important design decision is the degree of (de)normalization of the schema. The basic assumption is that content is mainly queried, so that creation and update cost is less important than lookup cost.

We consider two designs of denormalized schemas: materialization of reference sets and storage of relationships in a way that allows efficient queries. Efficient storage is based on the usage of numeric IDs to reference concepts and computing relationships based on ID sets. An example of such an approach is the BIRD numbering scheme for trees [13] that allows range queries to determine subtrees.

A. Storing Refinements

Compared to the straightforward conceptual model, the logical schema is denormalized in order to avoid repeated

navigation of specialization relationships when collecting the set of (transitive) base concepts or refinements of a concept.

Two approaches are investigated: aggregated data and transitive refinement relationships.

Aggregated data collects necessary information to avoid nested queries for refinements. All base concepts and all refinements are stored in an object representing the concept definition in a certain context. Context-dependent content is added in contextual concept amendments (s.a.) that are stored as part of the context hierarchy. These aggregate the definitions effective in all parent contexts.

The description objects additionally reference each other via original references.

Alternatively, just transitive refinement relationships are materialized for every concept in every context. This way, transitive refinements are directly available, and base concepts can be collected using a simple query.

Fig. 5 shows an example for the sample from Fig. 2. The dashed boxes show the transitive refinements per relevant context. Base concepts can be determined by queries.

For example, the (transitive) base concepts of *Concept 4* are those concepts that have this concept as a refinement. Specifically, these are *Concept 1* and *Concept 2* (in both the default context and in the context of *Concept 9*).

Storing the context together with the refinement relationships is vital for handling singleton (is the) relationships, in particular the unbinding of concepts.

B. Storing Context Hierarchies

Performance is particularly important for the retrieval of the hierarchy of contexts a concept is defined or amended in. The effective definition of a concept (including aggregated base concepts and content) relies on this concept hierarchy.

By blending in the context information into the transitive refinements, as shown in the previous subsection, the situation is leveraged to a large degree. Still, the content that a concept has in a certain context is also relevant to concept evaluations.

As for the specialization/generalization relationships, two approaches are discussed here: materialized content collections in all contexts and information about paths in the context hierarchy.

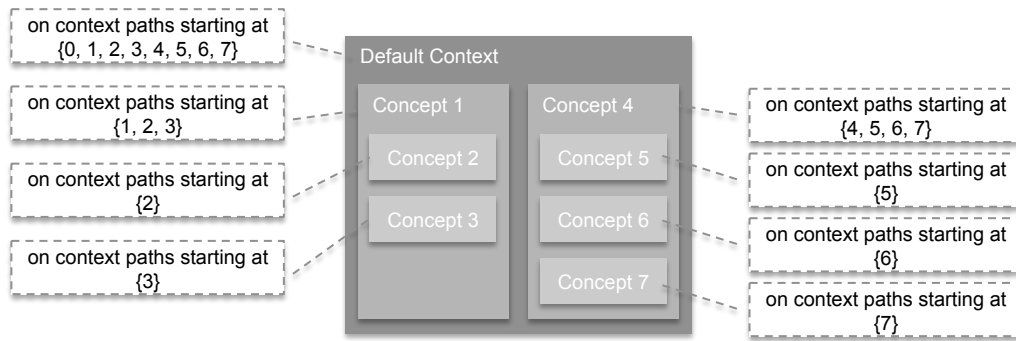


Figure 6. Representation of context hierarchies by materializing paths.

The materialization of contextual definitions works the same way as that of refinements: with every concept definition amendment, we store the effective content in the respective context. This has to be computed on definition.

For the second approach, Fig. 6 illustrates the attribution of paths to the schematic example of Fig. 3. For each concept, we note down the concepts lying on the path in the context hierarchy from the default context to a specific context. For example, *Concept 1* lies on the paths from the default context to itself, to *Concept 2*, and to *Concept 3*.

We used numeric IDs to reference the concept (with the ID 0 given to the pseudo-concept for the default concept). IDs have to be ordered from the default context to sub-contexts. By querying for all concepts on the path of a concept, ordered by ID, we retrieve the path to that concept.

VIII. PHYSICAL CONTENT STORAGE MODELS

This section briefly discusses some implementation approaches to context-aware content models using different data models. Specifically, we present the basics of a mapping to relational databases, to a document-oriented database, a content repository, and a graph database.

A. Mapping M3L to a Relational Database

There is a range of approaches to storing trees and graphs in relational databases [14]. On the basis of these, we add materialized transitive relationships as described above.

Relational tables for the transitive context hierarchy can be defined by statements like (with numeric type INT):

```
CREATE TABLE concept (id INT PRIMARY KEY);
CREATE TABLE paths (
  concept_id      INT REFERENCES concept(id),
  terminal_concept INT REFERENCES concept(id),
  PRIMARY KEY (concept_id, terminal_concept));
```

The table *concept* holds concepts (both initial definitions and amendments) with artificial IDs (other data is omitted here). The second table holds the path information as indicated in Fig. 6. *concept_id* refers to the concept, *terminal_concept* refers to the concept on whose path the concept lies.

Data stored this way can be queried by, e.g.,

```
SELECT c.* FROM concept c, paths p
WHERE c.id = p.concept_id
AND p.terminal_concept = i
ORDER BY p.concept_id DESC;
```

to retrieve the path to concept *i*.

Transitive refinements can be stored in a table:

```
CREATE TABLE transitive_refinements (
  base_concept_id INT REFERENCES concept(id),
  refinement_id    INT REFERENCES concept(id),
  context_id       INT REFERENCES concept(id),
  PRIMARY KEY (base_concept_id, refinement_id,
               context_id));
```

The base concepts of, e.g., *Concept 4* can be queried by:

```
SELECT base_concept_id
FROM transitive_refinements
WHERE refinement_id = 4 AND context_id = 0;
```

in the default context (with ID 0), or by:

```
SELECT base_concept_id
FROM transitive_refinements
WHERE refinement_id = 4 AND context_id = 9;
```

for the context of *Concept 9*.

B. MongoDB

As an example of so-called NoSQL approaches, we conduct ongoing experiments with MongoDB [15], a widely used document-oriented database management system.

The definition of concept relationships is done in a similar way as in relational databases: records have IDs, and records store IDs for references. There are no distinct relation structures, though. References are stored as document fields.

In contrast to a purely relational structure, documents allow representing nested contexts in a natural manner by embedded documents.

As an example of a schema, the *insert* statement shown in Fig. 7 stores the whole graph of Fig. 2.

This structure can be queried as required. For example, to find concepts with base concept *Concept 6* in the context of *Concept 9*, the *aggregate* statement in Fig. 7 can be applied.

C. Content Repository for Java Technology API (JCR)

In an attempt to define a content-specific database, the *Content Repository for Java Technology API (JCR)* standard has been set up in *Java Specification Requests JSR-170* [16] and *JSR-283* [17]. The standard is employed by some commercial content management system products.

The API implies a content model to be supported by JCR implementations. The data model behind JCR is similar to XML: It features hierarchies of *nodes*, where each node can have *properties*, attributes of one out of a set of predefined base types.

```

db.concept.insert({ name: "Default Context", content: [
  { name: "Concept 1", baseConcepts: null, content: null },
  { name: "Concept 2", baseConcepts: null, content: null },
  { name: "Concept 3", baseConcepts: null, content: null },
  { name: "Concept 4", baseConcepts: ["Concept 1", "Concept 2"], content: null },
  { name: "Concept 5", baseConcepts: ["Concept 2"], content: null },
  { name: "Concept 6", baseConcepts: ["Concept 3"], content: null },
  { name: "Concept 7", baseConcepts: ["Concept 4", "Concept 5"], content: null },
  { name: "Concept 8", baseConcepts: ["Concept 4", "Concept 5", "Concept 6"], content: null },
  { name: "Concept 9", baseConcepts: null, content: [
    { name: "Concept 7", baseConcepts: ["Concept 4", "Concept 5", "Concept 6"], content: null,
      original: "Concept 7" } ] } ] })
db.concept.aggregate([
  {$unwind:"$content"},{$replaceRoot:{newRoot:"$content"}},{$match:{name:"Concept 9"}},
  {$unwind:"$content"},{$replaceRoot:{newRoot:"$content"}},{$match:{baseConcepts:"Concept 6"}}])

```

Figure 7. Document definitions to map M3L to MongoDB and a sample query.

With these characteristics, M3L concept models can be mapped to JCR in a straightforward manner: nodes represent concepts, and concept relationships are expressed in the node hierarchy as well as by properties of type *reference*.

Context hierarchy in M3L is reflected by the node hierarchy in JCR. This way, the API allows direct access to context by `Node#getParent()` and access to content by `Node#getNodes()`.

Relationships to base concepts are represented by a (multi-valued) reference property *base-ref*.

Contextual concept amendments are represented as nodes on their own as outlined in Section VI.C. Nodes for amendments link to the node representing the definition they add to by a reference property *original*.

A semantic rule is represented by a reference from a concept to the one that is defined by its rule. To this end, rule concepts that are instantiated on rule application are stored outside of concept hierarchy.

With this mapping, a *M3LStore* can be expressed easily using the JCR API. Functions regarding the context hierarchy are directly reflected in the node hierarchy, base concept references are expressed by reference properties of nodes.

Only refinement relationships require special consideration for navigating base concept references against their direction. E.g., for transitive refinements, Java code like the following is included (with *c* a node representing the concept for which to compute the transitive refinement, *ctx* a node representing the context in which to evaluate it, *refinement* a set in which to collect nodes):

```

outer: for (Node c2 : allConcepts()) {
  Node[] baseConcepts
  = baseConceptsTransitive(session, c2, ctx);
  for (Node bc : baseConcepts)
    if (bc.getPath().equals(c.getPath())) {
      refinement.add(c2);
      continue outer;
    }
}

```

This code is the core of the *refinementTransitive()* method of a *M3LStore* for JCR.

D. Mapping M3L to a Graph Database

Graph database management systems [18] organize data as graphs of different types.

In this section, the DMBS *Neo4J* [19] is considered as a representative of graph database management systems. It allows data modelling using directed colored graphs with labelled nodes. Data manipulation and querying is performed using the language *Cypher* [20].

In *Neo4j*, we model M3L concepts as nodes. Following the conceptual model from Section VI, we introduce types (labels) *CONCEPT* and *CONCEPTAMENDMENT* for initial concept definitions and for conceptual content amendments. For each contextual definition, an explicit node with a label is created. Edges representing concept relationships are set to and from nodes representing concepts in specific contexts.

For the different concept relationships occurring in M3L models, we add edges of different types. To express context, we use an edge of kind *CONTEXT* from a node representing a concept to a node representing the context of that concept. The relationship between a refinement and its base concept is represented by an edge of kind *BASE*. We record a reference from a contextual concept amendment to the concept it is redefining using an *ORIGINAL* edge. The semantic rule of a concept is expressed by a *SEMANTICRULE* edge from the concept to the new concept the rule defines.

Fig. 8 shows a database resulting from the concept definitions in the example of Person entities from Section III. It is a screen shot taken from the tool *Neo4j Browser*.

The node color shows the label assigned to a node: green for initial concept definition, blue for conceptual amendment.

Cypher allows expressing transitivity directly, e.g., using the path `()-[:BASE*]-()` for $(7)^T$ (*base^T*). Therefore, the basic concept definitions and access functions can be mapped to Cypher in a straightforward way.

Root level concepts are defined by a simple CREATE directive:

```
CREATE (c:CONCEPT) SET c.name='concept name'
```

In the mapping examples in this section, italicized terms are placeholders for parameter values. In the create directive this is the name of the concept to be created.

Nested concepts are defined in a given context by:

```

MATCH (ctx{name: 'context's concept name'})
  -[:CONTEXT]->... (t)
WHERE NOT (t)-[:CONTEXT]->()
CREATE (c:CONCEPT) SET c.name='concept name'
CREATE (c)-[:CONTEXT]->(ctx)

```


A. Summary

This article lays out approaches to context-aware content management, in particular using the Minimalistic Meta Modeling Language (M3L). Semantics is given to content by rules that allow M3L concept evaluation.

The architecture of a current testbed implementation is presented. The architecture description concentrates on basic functions required for M3L concept evaluation in a data layer. Since content bases typically become large in data volume, persistency has to be provided by this data layer.

Though it is easily possible to map context representations to existing data management approaches, care has to be taken to enable efficient querying for M3L concept evaluation.

A logical schema for the representation of contextual content is presented that introduces first optimizations that are independent of the target data model and the database management systems used.

First sketches of implementations using different data models are conducted. These demonstrate the feasibility of concept persistence using these data models.

Representative technologies for each data model are used to present schemas that can serve as a starting point of the discussion and evaluation of M3L implementations.

B. Outlook

The work on data model mappings for M3L concept definitions is ongoing work. There is ample room for further optimizations of the relational database schema with respect to query execution. The mappings to other data models, document-oriented, tree, and graph databases, need elaboration before significant comparisons between these can be conducted.

The utilization of databases to support M3L concept evaluation is an open issue. Currently, base functions are implemented by database queries while the overall evaluation process is performed in a generic way by application code. Other functions required for concept evaluation may be implemented efficiently in certain database models. One example is the computation of candidate lists for narrowings(10) that may be formulated using database-specific queries.

Experiments with different implementations are ongoing. Data models have yet to be rated based on practical results. To this end, implementations need to be optimized.

For comparison, a kind of test suite needs to be defined. Models and rule sets that address realistic scenarios will guide the investigations in the future. Data of significant volume has to be generated as concept instances according to such models.

ACKNOWLEDGMENT

Though the ideas presented in this paper are in no way related to the work at Namics, the author is thankful to his employer for letting him follow his research ambitions based on experience made in customer projects.

Discussions with colleagues, partners, and customers are highly appreciated.

Thanks go to the reviewers of the original conference paper as well as to those of this journal article.

REFERENCES

- [1] H.-W. Sehring, "Schemas for Context-aware Content Storage," Proc. Tenth Int. Conference on Creative Content Technologies (CONTENT 2018), pp. 18-23, Sep. 2018.
- [2] M. Gutmann, "Information Technology and Society," Swiss Federal Institute of Technology Zurich / Ecole Centrale de Paris, 2001.
- [3] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca, "A Data-oriented Survey of Context Models," ACM SIGMOD Record, vol. 36, pp. 19-26, December 2007.
- [4] A. Zimmermann, M. Specht, and A. Lorenz, "Personalization and Context Management," User Modeling and User-Adapted Interaction, vol. 15, pp. 275-302, Aug. 2005.
- [5] S. Trullemans, L. Van Holsbeeke, and B. Signer, "The Context Modelling Toolkit: A Unified Multi-layered Context Modelling Approach," Proc. ACM Human-Computer Interaction (PACMHCI), vol. 1, June 2017, pp. 7:1-7:16.
- [6] G. Orsi and L. Tanca, "Context Modelling and Context-Aware Querying (Can Datalog Be of Help?)," Proc. First International Conference on Datalog Reloaded (Datalog '10), Mar. 2010, pp. 225-244.
- [7] D. Ayed, C. Taconet, and G. Bernard, "A Data Model for Context-aware Deployment of Component-based Applications onto Distributed Systems," GET/INT, 2004.
- [8] S. Vaupel, D. Wlochowicz, and G. Taentzer, "A Generic Architecture Supporting Context-Aware Data and Transaction Management for Mobile Applications," Proc. International Conference on Mobile Software Engineering and Systems (MOBILESoft '16), May 2016, pp. 111-122.
- [9] J. W. Schmidt and H.-W. Sehring, "Conceptual Content Modeling and Management," Perspectives of System Informatics, vol. 2890, M. Broy and A.V. Zamulin, Eds. Springer-Verlag, pp. 469-493, 2003.
- [10] H.-W. Sehring, "Content Modeling Based on Concepts in Contexts," Proc. Third Int. Conference on Creative Content Technologies (CONTENT 2011), pp. 18-23, Sep. 2011.
- [11] C. Diggins: *Explicit Structural Typing (Duck Typing)*. [Online]. Available from <http://www.drdoobs.com/architecture-and-design/explicit-structural-typing-duck-typing/228701413>
- [12] T. Parr, The Definitive ANTLR 4 Reference. Pragmatic Bookshelf, 2013.
- [13] F. Weigel, K. U. Schulz, and H. Meuss, "The BIRD Numbering Scheme for XML and Tree Databases – Deciding and Reconstructing Tree Relations using Efficient Arithmetic Operations," Proc. Third international conference on Database and XML Technologies (XSym'05), Aug. 2005, pp. 49-67.
- [14] V. Tropashko, SQL Design Patterns: The Expert Guide to SQL Programming. Rampant Techpress, 2006.
- [15] K. Chodorow and M. Dirolf, MongoDB: The Definitive Guide. O'Reilly Media, Inc., 2010.
- [16] Day Software AG: *Content Repository for Java Technology API Specification (JSR-170)*. [Online]. Available from <http://docs.adobe.com/content/docs/en/spec/jcr/1.0/index.html>
- [17] Oracle Corporation: *JSR 283: Content Repository for Java™ Technology API Version 2.0*. [Online]. Available from <https://jcp.org/en/jsr/detail?id=283>
- [18] R. Angles, M. Arenas, P. Barceló, A. Hogan, J. Reutter, and D. Vrgoč, "Foundations of Modern Query Languages for Graph Databases," in ACM Computing Surveys vol. 50, September 2017.
- [19] J. Baton, R. Van Bruggen, Learning Neo4j 3.x. Packt, 2017.
- [20] I. Robinson, J. Webber, and E. Eifrem, Graph Databases: New Opportunities for Connected Data. O'Reilly Media, 2015.

A Novel Training Algorithm based on Limited-Memory quasi-Newton Method with Nesterov's Accelerated Gradient in Neural Networks and its Application to Highly-Nonlinear Modeling of Microwave Circuit

Shahrzad MAHBOUBI[†] and Hiroshi NINOMIYA^{††}

Graduate school of Electrical and Information Engineering, Shonan Institute of Technology

Email: 18T2012@sit.shonan-it.ac.jp[†], ninomiya@info.shonan-it.ac.jp^{††}

Abstract—This paper describes a novel algorithm based on Limited-memory quasi-Newton method incorporating Nesterov's accelerated gradient for faster training of neural networks. Limited-memory quasi-Newton is one of the most efficient and practical algorithms for solving large-scale optimization problems. Limited-memory quasi-Newton is also the gradient-based algorithm using the limited curvature information without the approximated Hessian such as the normal quasi-Newton. Therefore, Limited-memory quasi-Newton attracts attention as the training algorithm for large-scale and complicated neural networks. On the other hand, Nesterov's accelerated gradient method has been widely utilized as the first-order training algorithm for neural networks. This method accelerated the steepest gradient method using the inertia term for the gradient vector. In this paper, it is confirmed that the inertia term is effective for the acceleration of Limited-memory quasi-Newton based training of neural networks. The acceleration of the proposed algorithm is demonstrated through the computer simulations compared with the conventional training algorithms for a benchmark problem and a real-world problem of the microwave circuit modeling.

Keywords—Neural networks; training algorithm; Limited-memory quasi-Newton method; Nesterov's accelerated gradient method; highly-nonlinear function modeling.

I. INTRODUCTION

This paper extends our previous work [1], presented at the IARIA FUTURE COMPUTING 2018, on acceleration of Limited-memory quasi-Newton based training of neural networks.

Neural networks have been recognized as a useful tool for function approximation problems. Especially, neural networks can efficiently approximate functions with highly-nonlinear input-output properties [2]. For example, neural networks can be utilized as the microwave circuit modeling in which the network is trained from Electro-Magnetic (EM) data over a range of geometrical parameters and trained networks become models providing fast solutions of the EM behavior it learned [3]-[6]. Generally, EM behaviors for geometrical behaviors are highly-nonlinear [3]. This is useful for modeling where formulas are not available or original models are computationally too expensive.

Training is the most important step in developing a neural network model. Gradient-based algorithms are popularly used for the training and can be divided into two categories: first-order methods and second or approximated second order methods [2]. The formers are popularly used for this purpose [8]-[14]. The typical first-order training is the steepest gradient descent method so-called Backpropagation (BP) [2]. BP was accelerated by the momentum (inertia) term [8]. This technique

was referred to as Classical Momentum method (CM). A simple modification to improve the performance of CM was introduced as Nesterov's Accelerated Gradient method (NAG) [7][8]. On the other hand, the training algorithms need strategies to determine *stepsize* or *learning rate* and the efficiency of training is highly dependent on the stepsize. Adaptive gradient method (AdaGrad) [9] and Resilient Mean Square backpropagation (RMSprop) [12] were introduced for the neural network training with the adaptive stepsize. Moreover, the combination algorithm of the momentum acceleration and the adaptive stepsize was Adam [14]. The recent developments of training were mostly based on the stochastic strategies such as the minibatch methods in which the gradients were calculated using the portion of all training samples. However, stochastic strategies are not suitable for the neural network training with highly-nonlinear properties [15]. Therefore, the full batch strategies are focused on this paper. Note that to the best of author's knowledge, the convergence for the non-convex problems such as the neural network training was only discussed for Adam of full batch strategies [16].

Adam is the most popular and effective first-order algorithm. With the progress of AI and IoT technologies, however, the characteristics between inputs and desired outputs of the training samples have become more complex. For such scenarios, neural networks need to deal with highly-nonlinear functions. Under such circumstances, the first order methods converge too slowly and optimization error cannot be effectively reduced within finite time in spite of its advantage [17]. The second and approximated second order methods are represented by Newton and quasi-Newton (QN) methods, respectively. Particularly, the QN training, which is one of the most effective optimization [18] is widely utilized as the robust training algorithm for highly-nonlinear function approximations [3]-[6]. However, the QN iteration includes the product matrix (the approximated Hessian) and vector, that is QN needs the massive computer resources of memories as the scale of neural network becomes larger. QN incorporating Limited-memory scheme so-called Limited-memory QN (LQN) is effective for solving large-scale problems whose Hessian matrices cannot be computed at a reasonable cost or is not sparse [18][19]. Furthermore, the momentum acceleration of QN was introduced as Nesterov's accelerated quasi-Newton method (NAQ) [17]. It was shown that the inertia term was effective to reduce the number of iterations of QN and to accelerate its convergence speed.

In this paper, the acceleration technique of LQN is proposed using Nesterov's accelerated gradient [1]. First of all,

a novel algorithm is derived from the detailed consideration of the derivation process of NAQ. The proposed algorithm, which is referred to as Limited-memory NAQ (LNAQ), is accelerated incorporating the momentum acceleration scheme of NAQ into LQN. The proposed training is demonstrated through the computer simulations. The effectiveness of the inertia term is confirmed by the comparison of LNAQ with LQN using a benchmark problem of highly-nonlinear function approximation. Finally, it is shown that the proposed algorithm is efficient and practical for the real-world problem of microwave circuit modeling.

The contents of this paper is structured as follows: Section II shows the related works. Section III Introduces the formulation of training and conventional gradient-based algorithms such as BP, CM, NAG, AdaGrad, RMSprop, Adam and LQN. Section VI proposes the novel algorithm - LNAQ, which is the acceleration method of LQN by introducing momentum coefficient. Section V provides simulation results in order to demonstrate the validity of the proposed LNAQ. Section VI concludes this paper and describes the future works.

II. RELATED WORK

Recently, the neural networks having deep and huge structures have attracted enormous research attentions in pattern recognition, computer vision, and speech recognition [20]. First-order techniques such as CM, NAG, Adagrad, RMSprop, and Adam [7]-[9][12][14], have been used mostly for training of deep neural networks. On the other hand, neural network techniques have been recognized as useful tool for modeling and design optimization problems in analog and microwave circuits design of CAD [3]-[6][21]-[43] in which their I/O characteristics are strongly nonlinear. For example, EM behavior modeling [3]-[6], analog integrated circuits (IC) [23]-[26], oscillation [27][28], antenna applications [29], nonlinear microwave circuit optimization [30]-[32], waveguide filters [33]-[36], low-pass filters [17][36]-[38][49], power amplifier modeling [39]-[42], vias and interconnects [43], have been studied. Neural networks can be used for developing new models whose formula are not available or original models are computationally too expensive. In these studies, the neural networks with deep structure are not necessarily utilized. Suitable training algorithms for these purposes are approximated second-order methods such as QN and Levenberg-Marquardt method (LM). These methods produce models with lower training error and have faster speed of training than first-order methods [5]. LM method is a modified version of the Gauss-Newton method (GN). Particularly, LM can be thought of as a combination of the strong convergence ability of Steepest Gradient method and the rapid convergence speed of GN [44]-[46]. However, LM needs to solve the system of linear equations in each iteration [19]. On the contrary QN iterates approximating the inverse matrix of Hessian [18][19]. Therefore, QN did not need the matrix solution in each iteration, but had to handle the variable-metric matrix. As a result, these algorithms were unsuitable for training large-scale neural networks with much small errors. That is, for modeling of large-scale problems with highly-nonlinearity, the matrix handling has not reasonable cost [18]. Therefore, LQN was used for the training [18][19]. The main idea of LQN is to use curvature information from only the most recent iterations to construct the Hessian approximation. Curvature information

from earlier iteration, which is less likely to be relevant to the actual behavior of the Hessian at the current iteration, is discarded in the interest of saving memory [18].

On the other hand, the acceleration algorithm of QN was proposed as NAQ in [17]. This method can have realized introducing momentum coefficient into QN and drastically improved the convergence speed of the QN. As far as the author's best knowledge, NAQ was the first acceleration technique of QN using the momentum term.

III. FORMULATION OF TRAINING AND GRADIENT-BASED TRAINING METHODS

A. Formulation of training

Let \mathbf{d}_p , \mathbf{o}_p , and $\mathbf{w} \in \mathbb{R}^D$ be the p -th desired, output, and weight vectors, respectively. The error function $E(\mathbf{w})$ is defined as the mean squared error (MSE) of

$$E(\mathbf{w}) = \frac{1}{|T_r|} \sum_{p \in T_r} E_p(\mathbf{w}), \quad E_p(\mathbf{w}) = \frac{1}{2} \|\mathbf{d}_p - \mathbf{o}_p\|^2, \quad (1)$$

where T_r denotes a training data set $\{\mathbf{x}_p, \mathbf{d}_p\}, p \in T_r$ and $|T_r|$ is the number of training samples. Among the gradient-based algorithms, (1) is minimized by

$$\mathbf{w}_{k+1} = \mathbf{w}_k + \mathbf{v}_{k+1}, \quad (2)$$

where k is the iteration count and \mathbf{v}_{k+1} is the update vector. A simple gradient descent algorithm that is the original BP [2] has

$$\mathbf{v}_{k+1} = -\alpha_k \nabla E(\mathbf{w}_k), \quad (3)$$

where α_k is the stepsize and $\nabla E(\mathbf{w}_k)$ is the gradient vector at \mathbf{w}_k .

B. First-order training methods

This section introduces the conventional first-order training methods.

1) Classical Momentum method (CM)

CM is a technique for accelerating BP that accumulates a previous update vector in directions of persistent reduction in the training [8]. The update vector of CM is given by

$$\mathbf{v}_{k+1} = \mu \mathbf{v}_k - \alpha_k \nabla E(\mathbf{w}_k), \quad (4)$$

where $0 \leq \mu \leq 1$ denotes the momentum coefficient.

2) Nesterov's Accelerated Gradient method (NAG)

NAG has been the subject of much recent studies in machine learning [7][8][17]. Arguing that NAG can be viewed as a simple modification of CM, and can sometimes provide a distinct improvement in performance for acceleration of neural network training [8]. CM computes the gradient vector from the current position \mathbf{w}_k , whereas NAG first performs a partial update to \mathbf{w}_k , computing $\mathbf{w}_k + \mu \mathbf{v}_k$, and then computes the

gradient at $\mathbf{w}_k + \mu \mathbf{v}_k$. NAG have better convergence rate than CM [7]. NAG update can be written as

$$\mathbf{v}_{k+1} = \mu_k \mathbf{v}_k - \alpha_k \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k), \quad (5)$$

where $\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$ means the gradient of $E(\mathbf{w})$ at $\mathbf{w}_k + \mu \mathbf{v}_k$ and is referred to as Nesterov's accelerated gradient vector.

3) Adaptive Gradient method (AdaGrad)

AdaGrad [9] is the first-order gradient-based training algorithms with an adaptive stepsize. The update vector of AdaGrad is given by

$$v_{k+1,i} = -\frac{\alpha}{\sqrt{\sum_{s=1}^k (\nabla E(\mathbf{w}_s)_i)^2}} \nabla E(\mathbf{w}_k)_i. \quad (6)$$

Here $v_{k+1,i}$ and $\nabla E(\mathbf{w}_k)_i$ are the i -th elements of \mathbf{v}_{k+1} and $\nabla E(\mathbf{w}_k)$, respectively. α is a global stepsize shared by all dimensions. The recommended value of α is $\alpha = 0.01$ [9][10][11].

4) Resilient Mean Square backpropagation method (RMSprop)

RMSprop [12] was a mini-batch version of Rprop [10]. The update vector of RMSprop is

$$v_{k+1,i} = -\frac{\alpha}{\sqrt{\theta_{k,i} + \lambda}} \nabla E(\mathbf{w}_k)_i, \quad (7)$$

where $\lambda = 10^{-8}$ and

$$\theta_{k,i} = \gamma \theta_{k-1,i} + (1 - \gamma) (\nabla E(\mathbf{w}_k)_i)^2. \quad (8)$$

$\theta_{k,i}$ is the parameter of k -th iteration and i -th element. γ and the global stepsize of α are set to 0.9 and 0.001, respectively in [12].

5) Adam

Adam is the most popular and effective gradient-based training algorithm with less memory requirement [14]. Adam was realized by combining RMSprop with CM. The update vector of Adam can be written as

$$v_{k+1,i} = -\alpha \frac{\hat{m}_{k,i}}{(\sqrt{\hat{\theta}_{i,k}} + \lambda)}, \quad (9)$$

where

$$\hat{m}_{k,i} = \frac{m_{i,k}}{(1 - \gamma_1^k)}, \quad (10)$$

and

$$\hat{\theta}_{k,i} = \frac{\theta_{k,i}}{(1 - \gamma_2^k)}. \quad (11)$$

Here, $m_{k,i}$ and $\theta_{k,i}$ are given by

$$m_{k,i} = \gamma_1 m_{k-1,i} + (1 - \gamma_1) \nabla E(\mathbf{w}_k)_i, \quad (12)$$

and

$$\theta_{k,i} = \gamma_2 \theta_{k-1,i} + (1 - \gamma_2) (\nabla E(\mathbf{w}_k)_i)^2, \quad (13)$$

where $\lambda = 10^{-8}$ and γ_1^k and γ_2^k denote the k -th power of γ_1 and γ_2 , respectively. α is the global stepsize and the recommended value is $\alpha = 0.001$ [14]. $m_{k,i}$ and $\theta_{k,i}$ are i -th elements of the gradient and the squared gradient, respectively. The hyper-parameters $0 \leq \gamma_1, \gamma_2 < 1$ control the exponential decay rates of these running averages. The running average themselves are estimates of the first (the mean) moment and the second raw (the uncentered variance) moment of the gradient. γ_1 and γ_2 are chosen to be 0.9 and 0.999, respectively in [14]. All operations on vectors are element-wise.

Note that the recent developments of the training algorithm such as AdaGrad, RMSprop and Adam were based on the stochastic strategies. These strategies are not suitable for the training of highly-nonlinear function modeling [5][15]. Therefore, we focus on the methods using the curvature information and the full batch strategy in this paper.

C. Limited-memory quasi-Newton method (LQN)

QN method is the efficient optimization algorithm using the curvature information and commonly used as training method for highly-nonlinear function problems. The update vector of QN is defined as

$$\mathbf{v}_{k+1} = \alpha_k \mathbf{c}_k, \quad (14)$$

where

$$\mathbf{c}_k = -\mathbf{H}_k \nabla E(\mathbf{w}_k), \quad (15)$$

\mathbf{c}_k is the direction vector and \mathbf{H}_k is a symmetric positive definite matrix. \mathbf{H}_k is iteratively given by the Broyden-Fletcher-Goldfarb-Shanno (BFGS) formula of (16) as the approximated inverse matrix of Hessian [18][19].

$$\mathbf{H}_{k+1} = \mathbf{H}_k - \frac{(\mathbf{H}_k \mathbf{y}_k) \mathbf{s}_k^T + \mathbf{s}_k (\mathbf{H}_k \mathbf{y}_k)^T}{\mathbf{s}_k^T \mathbf{y}_k} + \left(1 + \frac{\mathbf{y}_k^T \mathbf{H}_k \mathbf{y}_k}{\mathbf{s}_k^T \mathbf{y}_k} \right) \frac{\mathbf{s}_k \mathbf{s}_k^T}{\mathbf{s}_k^T \mathbf{y}_k}, \quad (16)$$

where

$$\mathbf{s}_k = \mathbf{w}_{k+1} - \mathbf{w}_k, \quad (17)$$

$$\mathbf{y}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k) + \xi_k \mathbf{s}_k = \epsilon_k + \xi_k \mathbf{s}_k, \quad (18)$$

and ξ_k is defined as

$$\xi_k = \omega \|\nabla E(\mathbf{w}_k)\| + \max\{-\epsilon_k^T \mathbf{s}_k / \|\mathbf{s}_k\|^2, 0\}, \quad (19)$$

$$\begin{cases} \omega = 2 & \text{if } \|\nabla E(\mathbf{w}_k)\|^2 > 10^{-2}, \\ \omega = 100 & \text{if } \|\nabla E(\mathbf{w}_k)\|^2 < 10^{-2}. \end{cases} \quad (20)$$

Here, ξ_k was introduced to guarantee the numerical stability and the global convergence [47]. For the purpose of reducing the amount of computer resources used in QN, a sophisticated technique incorporating the limited-memory scheme is widely utilized for the calculation of \mathbf{v}_{k+1} as LQN. Specifically, this method is useful for solving problems whose \mathbf{H}_k in (16) cannot be computed at a reasonable cost. That is, instead of storing $D \times D$ matrix of \mathbf{H}_k , only $2 \times t \times D$ elements have to be stored. Furthermore, the product of the matrix and vector can be changed to only the inner product of stored vectors. Here, D is the dimension of \mathbf{w} and $t (\ll D)$ is a hyper-parameter defined by user. That is, \mathbf{s}_i and \mathbf{y}_i vectors between $i = k$ and $i = k - t$ are stored in LQN. As a result, the computational resources of memory and calculation costs are drastically reduced when $t \ll D$ [18]. The LQN scheme is illustrated in Algorithms 1 and 2. In Algorithm 1, α_k is derived using the line search in which Armijo's condition of (21) is utilized.

$$E(\mathbf{w}_k + \alpha_k \mathbf{c}_k) \leq E(\mathbf{w}_k) + \chi \alpha_k \nabla E(\mathbf{w}_k)^T \mathbf{c}_k, \quad (21)$$

where $0 < \chi < 1$ and $\chi = 0.001$ in this paper.

Algorithm 1: Limited-memory quasi-Newton (LQN)

1. $k = 1$;
 2. $\mathbf{w}_1 = \text{rand}[-0.5, 0.5]$ (uniform random numbers);
 3. Calculate $\nabla E(\mathbf{w}_1)$;
 4. **While** ($k < k_{max}$)
 - (a) Calculate the direction vector \mathbf{c}_k using Algorithm 2;
 - (b) Calculate stepsize α_k using Armijo's condition;
 - (c) Update $\mathbf{w}_{k+1} = \mathbf{w}_k + \alpha_k \mathbf{c}_k$;
 - (d) Calculate $\nabla E(\mathbf{w}_{k+1})$;
 - (e) $k = k + 1$;
 5. **return** \mathbf{w}_k ;
-

Algorithm 2: Direction Vector of LQN

1. $\mathbf{c}_k = -\nabla E(\mathbf{w}_k)$;
 2. for $i : k, k - 1, \dots, k - \min(k, (t - 1))$;
 - (a) $\beta_i = \mathbf{s}_i^T \mathbf{c}_k / \mathbf{s}_i^T \mathbf{y}_i$;
 - (b) $\mathbf{c}_k = \mathbf{c}_k - \beta_i \mathbf{y}_i$;
 3. if $k > 1$, $\mathbf{c}_k = (\mathbf{s}_k^T \mathbf{y}_k / \mathbf{y}_k^T \mathbf{y}_k) \mathbf{c}_k$;
 4. for $i : k - \min(k, (t - 1)), \dots, k - 1, k$;
 - (a) $\tau = \mathbf{y}_i^T \mathbf{c}_k / \mathbf{y}_i^T \mathbf{s}_i$;
 - (b) $\mathbf{c}_k = \mathbf{c}_k - (\beta_i - \tau) \mathbf{s}_i$;
 5. **return** \mathbf{c}_k ;
-

IV. PROPOSED ALGORITHM - LIMITED-MEMORY NESTEROV'S ACCELERATED QUASI-NEWTON METHOD (LNAQ)

NAQ training was derived from the quadratic approximation of (1) around $\mathbf{w}_k + \mu \mathbf{v}_k$ whereas QN used the approximation of (1) around \mathbf{w}_k [17]. NAQ drastically improved the convergence speed of QN using the gradient vector at $\mathbf{w}_k + \mu \mathbf{v}_k$ of $\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$ called Nesterov's accelerated

gradient vector [7][17]. This means that the inertia term of $\mu \mathbf{v}_k$ was effective to accelerate the QN. First of all, the derivation of NAQ is briefly introduced as follows:

Let $\Delta \mathbf{w}$ be the vector $\Delta \mathbf{w} = \mathbf{w} - (\mathbf{w}_k + \mu_k \mathbf{v}_k)$, the quadratic approximation of (1) around $\mathbf{w}_k + \mu_k \mathbf{v}_k$ is defined as

$$E(\mathbf{w}) \simeq E(\mathbf{w}_k + \mu_k \mathbf{v}_k) + \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k)^T \Delta \mathbf{w} + \frac{1}{2} \Delta \mathbf{w}^T \nabla^2 E(\mathbf{w}_k + \mu_k \mathbf{v}_k) \Delta \mathbf{w}, \quad (22)$$

where $\nabla^2 E(\mathbf{w}_k + \mu_k \mathbf{v}_k)$ is Hessian of $E(\mathbf{w})$. The minimizer of this quadratic function is explicitly given by

$$\Delta \mathbf{w} = -\nabla^2 E(\mathbf{w}_k + \mu_k \mathbf{v}_k)^{-1} \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k). \quad (23)$$

Then the new iterate is defined as

$$\mathbf{w}_{k+1} = (\mathbf{w}_k + \mu_k \mathbf{v}_k) - \nabla^2 E(\mathbf{w}_k + \mu_k \mathbf{v}_k)^{-1} \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k). \quad (24)$$

This iteration is considered as Newton method with the momentum term $\mu \mathbf{v}_k$. Here Hessian $\nabla^2 E(\mathbf{w}_k + \mu \mathbf{v}_k)$ is approximated by $\hat{\mathbf{B}}_{k+1}$ and the rank-2 updating formula of this matrix is derived. Let \mathbf{p}_k and \mathbf{q}_k be

$$\mathbf{p}_k = \mathbf{w}_{k+1} - (\mathbf{w}_k + \mu \mathbf{v}_k), \quad (25)$$

$$\mathbf{q}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k), \quad (26)$$

and the secant condition is defined as

$$\mathbf{q}_k = \hat{\mathbf{B}}_{k+1} \mathbf{p}_k. \quad (27)$$

The suitable rank-2 updating formula for $\hat{\mathbf{B}}_{k+1}$ is derived as follows. The matrix $\hat{\mathbf{B}}_{k+1}$ is defined using arbitrary vectors \mathbf{t} and \mathbf{u} and constants a and b as

$$\hat{\mathbf{B}}_{k+1} = \hat{\mathbf{B}}_k + a \mathbf{t} \mathbf{t}^T + b \mathbf{u} \mathbf{u}^T. \quad (28)$$

Substitute (28) into the secant condition (27),

$$\mathbf{q}_k = (\hat{\mathbf{B}}_k + a \mathbf{t} \mathbf{t}^T + b \mathbf{u} \mathbf{u}^T) \mathbf{p}_k = \hat{\mathbf{B}}_k \mathbf{p}_k + a \mathbf{t} (\mathbf{t}^T \mathbf{p}_k) + b \mathbf{u} (\mathbf{u}^T \mathbf{p}_k). \quad (29)$$

Since $\mathbf{t}^T \mathbf{p}_k$ and $\mathbf{u}^T \mathbf{p}_k$ are scalars, both of conditions $\mathbf{t} = \mathbf{q}_k$ and $\mathbf{u} = -\hat{\mathbf{B}}_k \mathbf{p}_k$ are necessary to the secant condition of (27). Furthermore scalars a and b are given by $a (\mathbf{t}^T \mathbf{p}_k) = 1$ and $b (\mathbf{u}^T \mathbf{p}_k) = 1$, respectively. As a result, the rank-2 updating formula for NAQ is defined as

$$\hat{\mathbf{B}}_{k+1} = \hat{\mathbf{B}}_k + \frac{\mathbf{q}_k \mathbf{q}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} - \frac{\hat{\mathbf{B}}_k \mathbf{p}_k \mathbf{p}_k^T \hat{\mathbf{B}}_k}{\mathbf{p}_k^T \hat{\mathbf{B}}_k \mathbf{p}_k}. \quad (30)$$

Next, it is shown that $\hat{\mathbf{B}}_{k+1}$ of (30) is the symmetric positive definite matrix under the *Definition*: $\hat{\mathbf{B}}_k$ is the symmetric positive definite matrix. Here the following conditions are guaranteed for the above:

- (a): $\hat{\mathbf{B}}_{k+1}$ of (30) satisfies the secant condition $\mathbf{q}_k = \hat{\mathbf{B}}_{k+1} \mathbf{p}_k$.
- (b): If $\hat{\mathbf{B}}_k$ is symmetry, $\hat{\mathbf{B}}_{k+1}$ is also symmetry.
- (c): If $\hat{\mathbf{B}}_k$ is the positive definite matrix, $\hat{\mathbf{B}}_{k+1}$ is also the positive definite matrix.

Proof of (a):

From (30) the secant condition $\mathbf{q}_k = \hat{\mathbf{B}}_{k+1} \mathbf{p}_k$:

$$\begin{aligned} \hat{\mathbf{B}}_{k+1} \mathbf{p}_k &= \left(\hat{\mathbf{B}}_k + \frac{\mathbf{q}_k \mathbf{q}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} - \frac{\hat{\mathbf{B}}_k \mathbf{p}_k \mathbf{p}_k^T \hat{\mathbf{B}}_k}{\mathbf{p}_k^T \hat{\mathbf{B}}_k \mathbf{p}_k} \right) \mathbf{p}_k \\ &= \hat{\mathbf{B}}_k \mathbf{p}_k + \frac{\mathbf{q}_k \mathbf{q}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} \mathbf{p}_k - \frac{\hat{\mathbf{B}}_k \mathbf{p}_k \mathbf{p}_k^T \hat{\mathbf{B}}_k}{\mathbf{p}_k^T \hat{\mathbf{B}}_k \mathbf{p}_k} \mathbf{p}_k = \mathbf{q}_k \end{aligned} \quad (31)$$

□

Proof of (b): This is clear from (30).

□

Proof of (c):

First, $\mathbf{q}_k^T \mathbf{p}_k > 0$ will be shown. When the stepsize α_k is calculated by the exact line search, that is,

$$dE(\mathbf{w}_{k+1})/d\alpha_k = -\nabla E(\mathbf{w}_{k+1})^T \hat{\mathbf{H}}_k \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k) = 0. \quad (32)$$

As a result,

$$\mathbf{q}_k^T \mathbf{p}_k = \alpha_k \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)^T \hat{\mathbf{H}}_k \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k) > 0, \quad (33)$$

is derived. It is guaranteed in (33) that $\hat{\mathbf{H}}_k$ is the positive definite matrix because it is the inverse matrix of $\hat{\mathbf{B}}_k$, and $\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k) \neq \mathbf{0}$.

Second, the positive definiteness of $\hat{\mathbf{B}}_{k+1}$, that is, let $\mathbf{r} \neq \mathbf{0}$ be an arbitrary vector, $\mathbf{r}^T \hat{\mathbf{B}}_{k+1} \mathbf{r} > 0$ will be shown. Because $\hat{\mathbf{B}}_k$ is the positive definite matrix, it can be divided as $\hat{\mathbf{B}}_k = \mathbf{C} \mathbf{C}^T$ using an arbitrary non-singular matrix \mathbf{C} . Let $\mathbf{t} = \mathbf{C}^T \mathbf{r} (\neq \mathbf{0})$ and $\mathbf{u} = \mathbf{C}^T \mathbf{p}_k (\neq \mathbf{0})$, it is shown that

$$\mathbf{r}^T \hat{\mathbf{B}}_{k+1} \mathbf{r} = \frac{(\mathbf{t}^T \mathbf{t})(\mathbf{u}^T \mathbf{u}) - (\mathbf{t}^T \mathbf{u})^2}{\mathbf{u}^T \mathbf{u}} + \frac{(\mathbf{r}^T \mathbf{q}_k)^2}{\mathbf{q}_k^T \mathbf{p}_k} \geq 0, \quad (34)$$

with the Cauchy-Schwarz inequality [18] and the condition of (33). In (34) the equal condition is satisfied, if and only if $(\mathbf{t}^T \mathbf{t})(\mathbf{u}^T \mathbf{u}) - (\mathbf{t}^T \mathbf{u})^2 = 0$ and $\mathbf{r}^T \mathbf{q}_k = 0$. The former equation holds when $\mathbf{t} = \psi \mathbf{u}$ with the arbitrary scalar $\psi (\neq 0)$. When $\mathbf{t} = \psi \mathbf{u}$, then $\mathbf{r} = \psi \mathbf{p}_k$. Therefore, the later equation is transformed as $\mathbf{r}^T \mathbf{q}_k = \psi \mathbf{p}_k^T \mathbf{q}_k = 0$. This contradicts (33). Then the equal condition of (34) is not satisfied. As a result, $\hat{\mathbf{B}}_{k+1}$ holds $\mathbf{r}^T \hat{\mathbf{B}}_{k+1} \mathbf{r} > 0$, namely positive definiteness.

□

Applying the Sherman-Morrison-Woodbury formula [18] to (30), the update formula of the inverse Hessian approximation $\hat{\mathbf{H}}_{k+1} (= \hat{\mathbf{B}}_{k+1}^{-1})$ is given by

$$\begin{aligned} \hat{\mathbf{H}}_{k+1} &= \hat{\mathbf{H}}_k - \frac{(\hat{\mathbf{H}}_k \mathbf{q}_k) \mathbf{p}_k^T + \mathbf{p}_k (\hat{\mathbf{H}}_k \mathbf{q}_k)^T}{\mathbf{p}_k^T \mathbf{q}_k} \\ &\quad + \left(1 + \frac{\mathbf{q}_k^T \hat{\mathbf{H}}_k \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k} \right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}. \end{aligned} \quad (35)$$

From the above, it is confirmed that the NAQ has a similar convergence property with QN because $\hat{\mathbf{B}}_{k+1}$ updated by (35) holds symmetry and positive definiteness and $\hat{\mathbf{H}}_{k+1}$ is the inverse matrix of $\hat{\mathbf{B}}_{k+1}$. The update vector \mathbf{v}_{k+1} of NAQ can be obtained as follow.

$$\mathbf{v}_{k+1} = \mu_k \mathbf{v}_k + \alpha_k \hat{\mathbf{c}}_k, \quad (36)$$

$$\hat{\mathbf{c}}_k = -\hat{\mathbf{H}}_k \nabla E(\mathbf{w}_k + \mu_k \mathbf{v}_k). \quad (37)$$

The momentum coefficient of μ was usually selected from value close to 1 such as $\{0.8, 0.85, 0.9, 0.95\}$ and fixed during iteration [8][17].

The limited-memory scheme can be straightly applied to the update of (36) in NAQ. The detail of the limited memory scheme is derived as follows. In the first, the update formula of (35) is transformed as

$$\begin{aligned} \hat{\mathbf{H}}_{k+1} &= \hat{\mathbf{H}}_k - \frac{(\hat{\mathbf{H}}_k \mathbf{q}_k) \mathbf{p}_k^T + \mathbf{p}_k (\hat{\mathbf{H}}_k \mathbf{q}_k)^T}{\mathbf{p}_k^T \mathbf{q}_k} \\ &\quad + \left(1 + \frac{\mathbf{q}_k^T \hat{\mathbf{H}}_k \mathbf{q}_k}{\mathbf{p}_k^T \mathbf{q}_k} \right) \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \\ &= \left(\mathbf{I} - \frac{\mathbf{q}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \right)^T \hat{\mathbf{H}}_k \left(\mathbf{I} - \frac{\mathbf{q}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \right) + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \end{aligned} \quad (38)$$

$$= \hat{\mathbf{G}}_k^T \hat{\mathbf{H}}_k \hat{\mathbf{G}}_k + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}, \quad (39)$$

where

$$\hat{\mathbf{G}}_k = \left(\mathbf{I} - \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k} \right). \quad (40)$$

Then $\hat{\mathbf{H}}_k$ is given by

$$\hat{\mathbf{H}}_k = \hat{\mathbf{G}}_{k-1}^T \hat{\mathbf{H}}_{k-1} \hat{\mathbf{G}}_{k-1} + \frac{\mathbf{p}_{k-1} \mathbf{p}_{k-1}^T}{\mathbf{p}_{k-1}^T \mathbf{q}_{k-1}}. \quad (41)$$

Substitute (41) into (39),

$$\begin{aligned} \hat{\mathbf{H}}_{k+1} &= \hat{\mathbf{G}}_k^T \hat{\mathbf{G}}_{k-1}^T \hat{\mathbf{H}}_{k-1} \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k + \hat{\mathbf{G}}_k^T \frac{\mathbf{p}_{k-1} \mathbf{p}_{k-1}^T}{\mathbf{p}_{k-1}^T \mathbf{q}_{k-1}} \hat{\mathbf{G}}_k \\ &\quad + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}. \end{aligned} \quad (42)$$

By repeating this operation until $k = 1$, the update formula of $\hat{\mathbf{H}}_{k+1}$ is retransformed as

$$\begin{aligned}\hat{\mathbf{H}}_{k+1} &= (\hat{\mathbf{G}}_1 \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k)^T \hat{\mathbf{H}}_1 (\hat{\mathbf{G}}_1 \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k) \\ &+ (\hat{\mathbf{G}}_2 \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k)^T \frac{\mathbf{p}_1 \mathbf{p}_1^T}{\mathbf{p}_1^T \mathbf{q}_1} (\hat{\mathbf{G}}_2 \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k) + \dots \\ &+ (\hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k)^T \frac{\mathbf{p}_{k-2} \mathbf{p}_{k-2}^T}{\mathbf{p}_{k-2}^T \mathbf{q}_{k-2}} (\hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k) + \\ &\quad \hat{\mathbf{G}}_k^T \frac{\mathbf{p}_{k-1} \mathbf{p}_{k-1}^T}{\mathbf{p}_{k-1}^T \mathbf{q}_{k-1}} \hat{\mathbf{G}}_k + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k},\end{aligned}\quad (43)$$

where $\hat{\mathbf{H}}_1$ is an initial positive definite symmetric matrix. Since the inverse Hessian approximation $\hat{\mathbf{H}}_k$ will generally be dense, so that the cost of storing and manipulating it is prohibitive when the number of variables is large [18]. To circumvent this problem, we apply the limited-memory scheme of LQN with the user defined parameter of t to (43). The limited-memory formula of (43) between $k - th$ and $(k - t) - th$ iteration is derived as

$$\begin{aligned}\hat{\mathbf{H}}_{k+1} &= (\hat{\mathbf{G}}_{k-t+1} \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k)^T \hat{\mathbf{H}}_k^0 (\hat{\mathbf{G}}_{k-t+1} \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k) \\ &+ (\hat{\mathbf{G}}_{k-t+2} \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k)^T \frac{\mathbf{p}_{k-t+1} \mathbf{p}_{k-t+1}^T}{\mathbf{p}_{k-t+1}^T \mathbf{q}_{k-t+1}} (\hat{\mathbf{G}}_{k-t+2} \dots \hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k) \\ &+ \dots + (\hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k)^T \frac{\mathbf{p}_{k-2} \mathbf{p}_{k-2}^T}{\mathbf{p}_{k-2}^T \mathbf{q}_{k-2}} (\hat{\mathbf{G}}_{k-1} \hat{\mathbf{G}}_k) + \\ &\quad \hat{\mathbf{G}}_k^T \frac{\mathbf{p}_{k-1} \mathbf{p}_{k-1}^T}{\mathbf{p}_{k-1}^T \mathbf{q}_{k-1}} \hat{\mathbf{G}}_k + \frac{\mathbf{p}_k \mathbf{p}_k^T}{\mathbf{p}_k^T \mathbf{q}_k}.\end{aligned}\quad (44)$$

By substituting (44) into (37), the search vector $\hat{\mathbf{c}}_k$ of proposed LNAQ is calculated [1]. Here, $\hat{\mathbf{G}}_k$ is defined by the identity matrix and the inner products. Therefore, the search vector of LNAQ can be obtained by performing a sequence of inner products and vector summations of pairs $\{\mathbf{p}_i, \mathbf{q}_i\} \ i : k - t + 1, \dots, k - 1, k\}$. After the new iterate \mathbf{w}_{k+1} is computed, the oldest vector pair in the set of pairs $\{\mathbf{p}_i, \mathbf{q}_i\}$ is deleted and replaced by the new pairs $\{\mathbf{p}_k, \mathbf{q}_k\}$. As a result, we can derive a recursive procedure to compute $\hat{\mathbf{c}}_k$. The LNAQ scheme is illustrated in Algorithms 3 and 4. Here, Armijo's condition of (45) for LNAQ is used for the line search.

$$E(\mathbf{w}_k + \mu \mathbf{v}_k + \alpha_k \hat{\mathbf{c}}_k) \leq E(\mathbf{w}_k + \mu \mathbf{v}_k) + \hat{\chi} \alpha_k \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)^T \hat{\mathbf{c}}_k, \quad (45)$$

where $0 < \hat{\chi} < 1$ and $\hat{\chi} = 0.001$ in this paper. Furthermore, in order to guarantees the numerical stability and the global convergence of LNAQ, (46) and (47) are added to \mathbf{q}_k similarly to LQN [47].

$$\hat{\xi}_k = \omega \|\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)\| + \max\{-\epsilon_k^T \mathbf{p}_k / \|\mathbf{p}_k\|^2, 0\}, \quad (46)$$

and

$$\begin{cases} \omega = 2 & \text{if } \|\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)\|^2 > 10^{-2}, \\ \omega = 100 & \text{if } \|\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)\|^2 < 10^{-2}. \end{cases} \quad (47)$$

As a result, \mathbf{q}_k is rewritten as

$$\mathbf{q}_k = \nabla E(\mathbf{w}_{k+1}) - \nabla E(\mathbf{w}_k + \mu \mathbf{v}_k) + \hat{\xi}_k \mathbf{p}_k = \epsilon_k + \hat{\xi}_k \mathbf{p}_k. \quad (48)$$

Algorithm 3: The proposed LNAQ

1. $k = 1$;
 2. $\mathbf{w}_1 = \text{rand}[-0.5, 0.5]$ (uniform random numbers);
 3. **While** ($k < k_{max}$)
 - (a) Calculate $\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$;
 - (b) Calculate the direction vector $\hat{\mathbf{c}}_k$ using Algorithm 4;
 - (c) Calculate stepsize α_k using Armijo's condition;
 - (d) Update $\mathbf{w}_{k+1} = \mathbf{w}_k + \mu \mathbf{v}_k + \alpha_k \hat{\mathbf{c}}_k$;
 - (e) Calculate $\nabla E(\mathbf{w}_{k+1})$;
 - (f) $k = k + 1$;
 4. **return** \mathbf{w}_k ;
-

Algorithm 4: Direction Vector of LNAQ

1. $\hat{\mathbf{c}}_k = -\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$;
 2. for $i : k, k - 1, \dots, k - \min(k, (t - 1))$;
 - (a) $\hat{\beta}_i = \mathbf{p}_i^T \hat{\mathbf{c}}_k / \mathbf{p}_i^T \mathbf{q}_i$;
 - (b) $\hat{\mathbf{c}}_k = \hat{\mathbf{c}}_k - \hat{\beta}_i \mathbf{q}_i$;
 3. if $k > 1$, $\hat{\mathbf{c}}_k = (\mathbf{p}_k^T \mathbf{q}_k / \mathbf{q}_k^T \mathbf{q}_k) \hat{\mathbf{c}}_k$;
 4. for $i : k - \min(k, (t - 1)), \dots, k - 1, k$;
 - (a) $\hat{\tau} = \mathbf{q}_i^T \hat{\mathbf{c}}_k / \mathbf{q}_i^T \mathbf{p}_i$;
 - (b) $\hat{\mathbf{c}}_k = \hat{\mathbf{c}}_k - (\hat{\beta}_i - \hat{\tau}) \mathbf{p}_i$;
 5. **return** $\hat{\mathbf{c}}_k$;
-

In Algorithm 3, two times calculations of the gradient vectors of $\nabla E(\mathbf{w}_k + \mu \mathbf{v}_k)$ and $\nabla E(\mathbf{w}_{k+1})$ were needed within a training loop whereas LQN needs one derivation of the gradient. This is a disadvantage of LNAQ, but the algorithm can further shorten the iteration counts to cancel out the effect of this shortcoming [1]. The simulation results will show the above fact.

V. SIMULATION RESULTS

Computer simulations are conducted in order to demonstrate the validity of the proposed LNAQ. In the simulations the feedforward neural networks with a hidden layer and an arbitrary number of hidden layer's neurons were used. Each neuron has a sigmoid function as $\text{sig}(x) = 1/(1 + \exp(-x))$. The performance of LNAQ is compared with conventional algorithms such as BP [2], CM [8], NAG [8], AdaGrad [9], RMSprop [12], Adam [14] and LQN [18] for two benchmark problems. Benchmark problems used here are a function approximation problem of Levy function [48] and a microwave circuit modeling problem of low-pass filter [17][36]-[38][49]. Ten independent runs were performed with different starting values of \mathbf{w} , which are initialized by uniform random numbers within $[-0.5, 0.5]$. Each hyper-parameter of AdaGrad, RMSprop and Adam is set to the default value of each original

paper, respectively. These adaptive methods are mainly utilized in the stochastic (mini-batch) mode. However, the problems in this paper need the full batch method [15]. Therefore, the full batch scheme is applied to all algorithms. The momentum coefficient of μ used in CM, NAG and LNAQ are 0.8, 0.85, 0.9 and 0.95 as [8][17]. The simulations were performed on the computer, which has Intel Core i7-8700 3.2GHz processor and 8GB memory. Each trained neural network was estimated by the average, best and worst of $E(\mathbf{w})$, the average of computational time (s) and the average of iteration counts (k). Each element of the input and desired vectors of T_r is normalized within $[-1.0, 1.0]$ in the simulations.

A. Levy function approximation problem

Levy function ($\mathbb{R}^n \rightarrow \mathbb{R}^1$) shown in (49) is used for the first function approximation problem. The Levy function is a multimodal function with highly-nonlinear characteristic. Therefore, the function usually used as a benchmark problem for the multimodal function optimization [48].

$$f(x_1 \dots x_n) = \frac{\pi}{n} \left\{ \sum_{i=1}^{n-1} [(x_i - 1)^2 (1 + 10 \sin^2(\pi x_{i+1}))] + 10 \sin^2(\pi x_1) + (x_n - 1)^2 \right\}, x_i \in [-4, 4], \forall i, \quad (49)$$

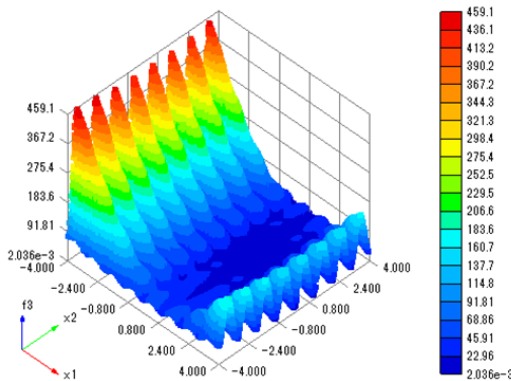


Figure 1. Levy Function $f(x_1, x_2)$.

where n denotes the input dimension. In Figure 1, Levy function with $n = 2$ dimensions of (49) is shown. From the figure, we can obtain the highly-nonlinear characteristic of the function. In this simulation the dimension of input vector \mathbf{x} is set to $n = 5$. The inputs and an output are x_1, \dots, x_5 and $f(x_1, \dots, x_5)$, respectively. The trained network has a hidden layer with 50 hidden neurons. Therefore, the structure of neural network is 5-50-1 and the dimension of \mathbf{w} is 351. The number of training data is $|T_r| = 5000$, which are generated by uniformly random number in $x_i \in [-4, 4]$. Maximum number of iteration is set to $k_{max} = 2 \times 10^4$. Here, we verified LNAQ from the viewpoints of two kinds of comparisons. First one is the comparison with LQN for iteration and computer time. Second, the proposed LNAQ is compared with the conventional algorithms for the training

errors.

1) Comparison of LQN and LNAQ

Here, we compare LNAQ and LQN with respect to iteration count (k) and the computational time (s) for several memories. The range of storage memory t is from 10 to 100 at intervals of 10. The terminate conditions are set to $E(\mathbf{w}) \leq 1.0 \times 10^{-4}$. For function approximation problems, the small MSE of $E(\mathbf{w})$ is very important, because the trained network with the small $E(\mathbf{w})$ can become an accurate neural network model. Therefore, the average iteration counts and computational times until $E(\mathbf{w}) \leq 1.0 \times 10^{-4}$ within $k_{max} = 2 \times 10^4$ are obtained

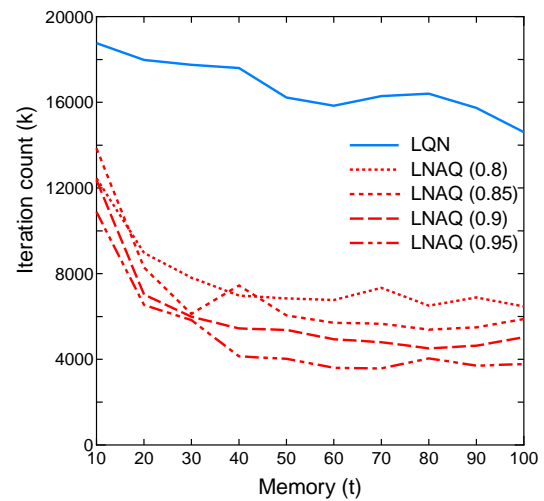


Figure 2. The average of iteration count vs memories.

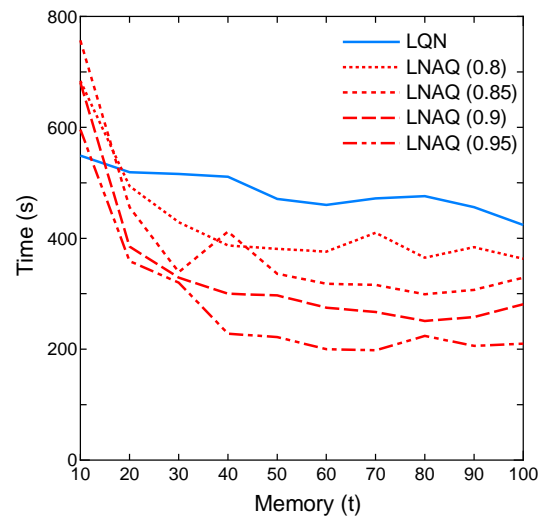
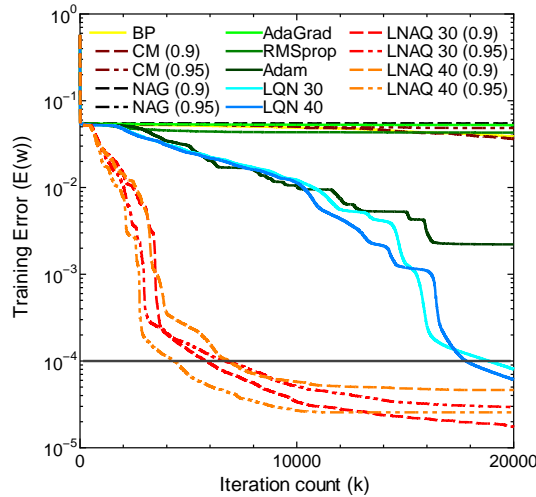


Figure 3. The average of calculation time vs memories.

in this comparison. Figure 2 shows the average of iteration count (k) vs memories (t) for LQN and LNAQ. From this figure, it can be seen that LNAQ converges with less iteration count than LQN regardless of μ . Therefore, LNAQ has the

TABLE I. Summary of Levy function.

Algorithm	μ	Memory	$E(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst	Time (s)	Time / Iteration (ms)
BP	-	-	38.8 / 30.6 / 52.0	487	24.35
CM	0.9	-	32.4 / 0.31 / 54.8	806	40.30
	0.95	-	47.5 / 13.4 / 54.8	854	42.70
NAG	0.9	-	55.0 / 54.8 / 55.7	802	40.10
	0.95	-	55.0 / 54.8 / 56.2	738	36.90
AdaGrad	-	-	52.5 / 52.3 / 53.1	488	24.40
RMSprop	-	-	43.0 / 33.2 / 54.1	487	24.35
Adam	-	-	1.20 / 0.16 / 10.2	487	24.35
LQN	-	30	0.0710 / 0.0338 / 0.167	732	36.60
	-	40	0.0679 / 0.0302 / 0.143	732	36.60
LNAQ	0.9	30	0.0174 / 0.00517 / 0.0448	1,210	60.50
	0.95	30	0.0295 / 0.00578 / 0.145	1,230	61.50
LNAQ	0.9	40	0.0205 / 0.00529 / 0.0407	1,220	61.00
	0.95	40	0.0256 / 0.00585 / 0.0583	1,250	62.50

Figure 4. The average training errors *vs* iteration count of Levy function.

ability to significantly reduce the iteration counts compared to LQN. Furthermore, it is shown that the decrease of iteration is hardly seen in memory (t) larger than 40 and the momentum coefficients μ closer to 1, that is $\mu = 0.9$ and 0.95 converge faster. However, LNAQ requires two calculations of gradient in one iteration. That is, it takes time to compare with LQN in one iteration. Therefore, it is necessary to compare the calculation time until the training end. Figure 3 shows the average of calculation times (s) *vs* memories (t) for LQN and LNAQ. From Figure 3, it can be seen that LNAQ is inferior to LQN in terms of calculation time, when the storage memory of (t) is small. However, when t increases, it is easy to conclude that LNAQ is faster than LQN. From these figures, it is confirmed that memories of $t = 30$ or 40 and coefficients of $\mu = 0.9$ and 0.95 are recommended.

2) Comparison of LNAQ and conventional algorithms

In these simulations, the proposed LNAQ is compared with BP, CM, NAG, AdaGrad, RMSprop, Adam and LQN. The storage amount of memories is experimentally set to $t = 30$ and 40 from the above results. The momentum coefficients μ of CM, NAG and LNAQ are set to $\mu = 0.9$ and 0.95. Here, the terminate condition is set to $k_{max} = 2 \times 10^4$. This means that the iteration continues after $E(\mathbf{w}) \leq 1.0 \times 10^{-4}$. The summary of results is shown in Table 1 and the average of training errors of BP, CM, NAG, AdaGrad, RMSprop, Adam, LQN and LNAQ for the iteration count is illustrated in Figure 4. From Figure 4 and Table 1, The conventional algorithms based on the first order methods such as BP, CM, NAG, AdaGrad and RMSprop could not converge to small training errors. From Table 1, it is confirmed that Adam, LQN and LNAQ converge to small errors depending on the initial value. In comparing of the average training errors, LQN and LNAQ can obtain the small average errors compared with Adam. Especially, the average error of LNAQ ($t = 30, \mu = 0.9$) is smallest and almost the same as the worst error. These results show the robustness with respect to the initial value. On the other hand, the calculation time of LQN is faster than the one of LNAQ for the same iteration count ($k_{max} = 2 \times 10^4$). This is caused by the drawback namely two times calculations of the gradients of LNAQ. However, LNAQ can reach the small training error ($E(\mathbf{w}) \leq 1.0 \times 10^{-4}$) faster than LQN. This fact can be confirmed in Figure 4.

B. Microwave circuit modeling of low-pass filter

Neural networks can be trained using measured or simulated microwave device data such as EM and physical data [3]-[6]. The trained neural networks can be used as models of microwave devices in place of CPU-intensive EM/physics models to significantly speed up circuit design while maintaining EM/physics-level accuracies [3][5]. Neural network based modeling has been used to model a variety of microwave circuit components at both device and circuit levels. In this simulation, we applied LNAQ to develop a neural network model of the microstrip low-pass filter (LPF) [17][36]-[38] illustrated in Figure 5. The dielectric constant and height of

TABLE II. Summary of LPF.

Algorithm	μ	Memory	$E(\mathbf{w})(\times 10^{-3})$ Ave/Best/Worst	Time (s)	Time/ Iteration (ms)
BP	-	-	22.4 / 19.6 / 24.3	143	2.86
CM	0.9	-	23.7 / 9.56 / 29.2	264	5.28
	0.95	-	24.0 / 6.41 / 29.2	277	5.54
NAG	0.9	-	106 / 16.3 / 832	248	4.96
	0.95	-	26.2 / 15.5 / 29.2	247	4.94
AdaGrad	-	-	25.3 / 24.8 / 25.6	142	2.84
RMSprop	-	-	26.6 / 26.2 / 27.0	144	2.88
Adam	-	-	5.54 / 4.66 / 6.35	142	2.84
LQN	-	30	6.89 / 5.81 / 7.68	246	4.92
	-	40	7.08 / 6.42 / 7.81	247	4.94
LNAQ	0.9	30	2.15 / 1.59 / 3.13	378	7.56
	0.95	30	1.63 / 1.36 / 2.06	377	7.54
LNAQ	0.9	40	1.97 / 1.57 / 2.80	380	7.60
	0.95	40	1.49 / 1.23 / 1.93	377	7.54

the substrate of LPF are 9.3mm and 1mm, respectively. The length D ranges 12-20mm at intervals of 1mm. The frequency range was 0.1 to 4.5GHz. Each set of contains 221 samples. The inputs of the neural network, x_1 and x_2 are frequency f and length D in which training data T_r and test data T_e are set to $D = [12, 14, 16, 18, 20]$ mm and $[13, 15, 17, 19]$ mm, respectively. The outputs, o_1 and o_2 are the magnitudes of S -parameters, $|S_{11}|$ and $|S_{21}|$, respectively. These data are obtaining by the standard software of *sonnet* [49]. Training data is illustrated in Figure 6. As shown in Figure 6, there are many irregularly aligned poles in S -parameters and the modeling of the poles is the most important in microwave circuit problems. Therefore the microwave circuit modeling is a strong nonlinear problem and needs very small training and testing errors. The number of hidden neurons is 45. Therefore, the structure of neural network is 2-45-2 and the dimension of \mathbf{w} is 227. The maximum iteration count is set to $k_{max} = 5 \times 10^4$. The purposed LNAQ is also compared with BP, CM, NAG, AdaGrad, RMSprop, ADAM and LQN. Memories (t) are selected 30 and 40 for both of LQN and LNAQ. The coefficients of μ for CM, NAG and LNAQ are set to 0.9 and 0.95. The summary of results is shown in Table 2 and the training errors for iteration counts are illustrated in Figure 7. From Table 2 and Figure 7, the first-order methods such as BP, CM, NAG, AdaGrad and RMSprop could not obtain the small training errors for the practical methods. Adam can obtain the small training errors compared with LQN for this problem. In comparison of Adam with LNAQ, LNAQ need more computational time than Adam because of two calculations of gradient and the complex procedure for the calculation of the direction. However, the proposed LNAQ can converge to smaller value of training error than Adam and LQN. Especially, the average training error of LNAQ ($t = 40$ and $\mu = 0.95$) can converge to 1.49×10^{-3} . Furthermore, LNAQ can obtain the small difference between the best and the worst errors. This means that LNAQ is also robust with respect to the initial value for this problem.

For measuring accuracy of modeling, the outputs of the

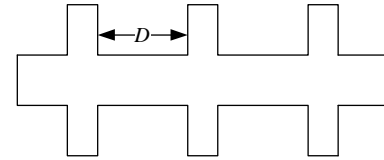


Figure 5. Layout of LPF.

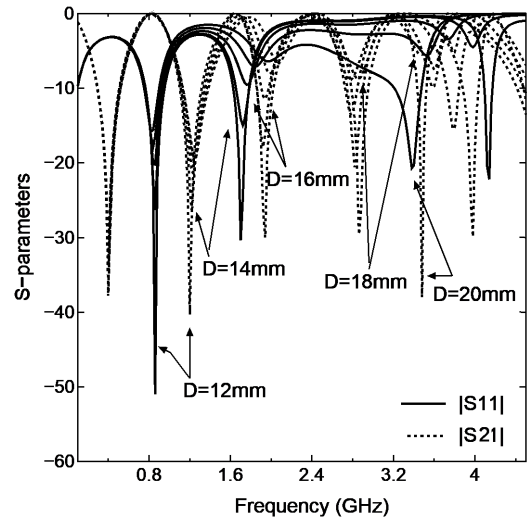


Figure 6. Training data set for LPF.

trained neural model for $D = [13, 15, 17, 19]$ mm, which are not included in training are compared with the test data of $|S_{11}|$ and $|S_{21}|$. The trained models are selected from neural networks trained by Adam and LNAQ ($t = 40$ and $\mu = 0.95$) with the smallest training errors 4.66×10^{-3} and 1.23×10^{-3} , respectively. The test errors $E_{test}(\mathbf{w})$ obtained by Adam and LNAQ are 3.15×10^{-3} and 0.656×10^{-3} , respectively. Figure 8 and 9 shows the comparison of the test data of $D = 13$

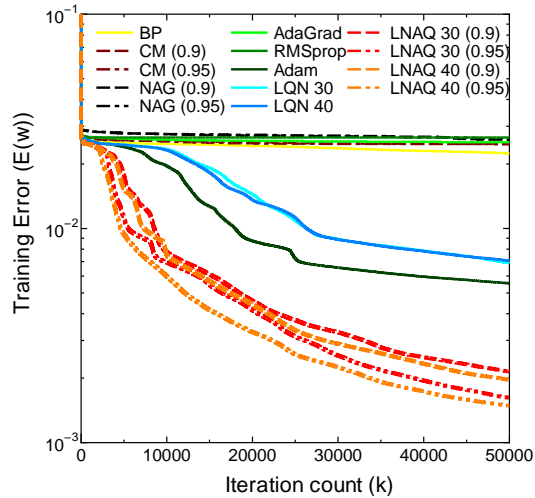


Figure 7. The average training errors *vs* iteration count of LPF.

mm and 17 mm with the model outputs trained by Adam and LNAQ, respectively. It can be seen from Figure 8 of the model trained by Adam that there are multiple large gaps between the model outputs and test data. They are prominent in places of pole. On the other hand, it can be confirmed that the neural model trained by LNAQ and the test data are showing good match between them from Figure 9.

VI. CONCLUSION

In this paper, we proposed a novel training algorithm called LNAQ, which was developed based on Limited-memory method of QN incorporating the momentum acceleration scheme and Nesterov's accelerated gradients vector. The effectiveness of the proposed LNAQ was demonstrated through the computer simulations compared with the conventional algorithms such as BP, CM, NAG, AdaGrad, RMSprop, Adam and LQN. For highly-nonlinear problem, the first-order methods such as BP, CM, NAG, AdaGrad and RMSprop could not obtain desired small training errors for the function approximation and the microwave circuit modeling problems. Only Adam could get small errors depending on the problem and the initial value of w . On the other hand, the curvature information-based method such as LQN and LNAQ could obtain the small errors for a function approximation problem. For a real-world problem of microwave circuit modeling the efficient and practical models could be trained by only LNAQ. Furthermore, the effectiveness of the momentum coefficient for QN with the limited memory scheme was demonstrated through the results of the training errors for iteration. This means that LNAQ can reduce training errors earlier than other method. LNAQ may take time to obtain a solution compared to other methods because of its drawback. Depending on the problem, however, it may be the only algorithm that can get a practical model that cannot be obtained by the other methods. This is very important issue for modeling of highly-nonlinear problems.

In the future the momentum parameters μ will be studied. This parameter was analytically determined for the first-order method of NAG in [7] for the convex problems whereas the fixed values were used in [8][16] for the neural training problems of the non-convex problems in the same way as

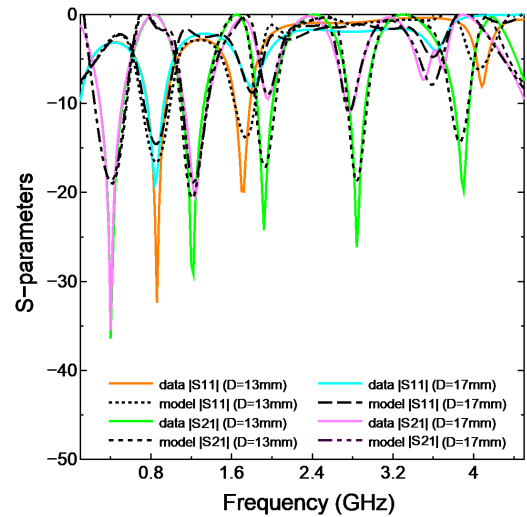


Figure 8. Example of comparison between test data and neural model trained by Adam.

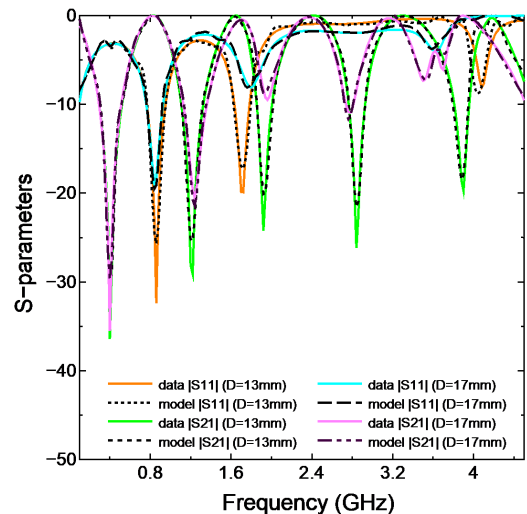


Figure 9. Example of comparison between test data and neural model trained by LNAQ.

this paper. Therefore, the analytical studies of the momentum parameter for the second-order method of LNAQ will be done in the future. Furthermore, the validity of the proposed algorithm for more highly-nonlinear function approximation problems and the much huge scale problems including deep networks will be demonstrated.

ACKNOWLEDGMENT

The authors thank to Prof. Q.J. Zhang at Carleton University, Canada, for his support of microwave circuit models. This work was supported by Japan Society for the Promotion of Science (JSPS), KAKENHI (17K00350).

REFERENCES

- [1] S. Mahboubi and H. Ninomiya, "A novel training algorithm based on limited-memory quasi-Newton method with Nesterov's accelerated gradient for neural networks," *IARIA / FUTURE COMPUTING'18*, pp. 1–3, Feb. 2018.

- [2] S. Haykin, "Neural Networks and Learning Machines 3rd," Pearson, 2009.
- [3] Q. J. Zhang, K. C. Gupta, and V. K. Devabhaktuni, "Artificial neural networks for RF and microwave design-from theory to practice," *IEEE Trans. Microwave Theory and Tech.*, vol.51, pp.1339–1350, Apr. 2003.
- [4] H. Ninomiya, "A hybrid global/local optimization technique for robust training and its application to microwave neural network models," *J.Signal Processing*, 14, 3, pp.213–222, 2010.
- [5] H. Kabir, L. Zhang, M. Yu, P. H. Aaen, J. Wood, and Q. J. Zhang, "Smart modeling of microwave devices," *IEEE Microwave Magazine*, vol.11, no.3, pp.105–118, May. 2010.
- [6] H. Ninomiya, S. Wan, H. Kabir, X. Zhang and Q. J. Zhang, "Robust training of microwave neural network models using combined global/local optimization techniques," *IEEE MTT-S International Microwave Symposium (IMS) Digest*, pp.995–998, Jun, 2008.
- [7] Y. Nesterov, "Introductory Lectures on Convex Optimization: A Basic Course," 2004.
- [8] I. Sutskever, J. Martens, G. Dahl, and G. Hinton, "On the importance of initialization and momentum in deep learning," *ICML'13*, 2013.
- [9] D. John, H. Elad, and S.Yoram, "Adaptive subgradient methods for online learning and stochastic optimization," *The Journal of Machine Learning Research*, pp.2121–2159, Jul. 2011.
- [10] S. Ruder, "An overview of gradient descent optimization algorithm," arXiv preprint arXiv:1609.04747. 2016.
- [11] M. D. Zeiler, "ADADELTA: An adaptive learning rate method," arXiv preprint arXiv:1212.5701. 2012.
- [12] T. Tieleman and G. Hinton, "Lecture 6.5 - RMSProp," *COURSERA: Neural Networks for Machine Learning. Technical report*, 2012.
- [13] M. Riedmiller and H. Braun, "Rprop - A fast adaptive learning algorithm," *ISCIS VII*, 1992.
- [14] P. D. Kinguma and J. Ba, "Adam: A method for stochastic optimization," *ICLR'15*, vol.5, May. 2015.
- [15] H. Ninomiya, "Dynamic sample size selection based quasi-Newton training for highly nonlinear function approximation using multilayer neural networks," *IEEE&INNS IJCNN'13*, pp.1932–1937, Aug. 2013.
- [16] A. Basu, S. De, A. Mukherjee, and E. Ullah, "Convergence guarantees for RMSProp and ADAM in non-convex optimization and their comparison to Nesterov acceleration on autoencoders," arXiv:1807.06766v1, Jul. 2018.
- [17] H. Ninomiya, "A novel quasi-Newton optimization for neural network training incorporating Nesterov's accelerated gradient," *IEICE NOLTA Journal*, vol.E8-N, no.4, pp.289–301, Oct. 2017.
- [18] J. Nocedal and S. J. Wright, "Numerical Optimization Second Edition," Springer, 2006.
- [19] W. Forst and D. Hoffmann, "Optimization - Theory and Practice," Springer, 2010.
- [20] I. Goodfellow, Y. Bengio, and A. Courville. "Deep learning (adaptive computation and machine learning series)," Adaptive Computation and Machine Learning series, 2016.
- [21] J. Wood and D. E. Root, eds, "Fundamentals of Nonlinear Behavioral Modeling for RF and Microwave Design," ARTECH HOUSE, 2005.
- [22] Q. J. Zhang, J. Bandler, S. Koziel H. Kabir, and L. Zhang, "ANN and space mapping for microwave modeling and optimization," *IEEE MTT-S International*, pp.980–983, May. 2010.
- [23] R. M. Hassani, D. Haerle, and R. Grosu, "Efficient modeling of complex analog integrated circuits using neural networks," *IEEE PRIME'12*, pp.1–4, Jun. 2016.
- [24] J. Michel and Y. Herve, "VHDL-AMS behavioral model of an analog neural networks based on a fully parallel weight perturbation algorithm using incremental on-chip learning," *IEEE International Symposium on Industrial Electronics*, vol.1, pp. 211–216, May. 2004.
- [25] A. Jafari, S. Sadri, and M. Zekri, "Design optimization of analog integrated circuits by using artificial neural networks," *IEEE SoCPar*, Nov. 2010.
- [26] R. M. Hasani, D. Haerle, C. F. Baumgartner, A. R. Lomuscio, and R. Grosu, "Compositional neural-network modeling of complex analog circuits," *IEEE IJCNN'17*, pp.2235–2242, May. 2017.
- [27] M. Kraemer, D. Dragomirescu, and Robert Plana, "A novel technique to create behavioral models of differential oscillators in VHDL-AMS," *SM2ACD*, Oct. 2008.
- [28] M. Kraemer, D. Dragomirescu, and R. Plana, "Nonlinear behavioral modeling of oscillators in VHDL-AMS using artificial neural networks," *IEEE RFIC*, pp.689–692, Jun. 2008.
- [29] J. P. Garcia, F. Q. Pereira, D. C. Rebenague, J. L. G. Tornero, and A. A. Melcon, "A neural-network method for the analysis of multilayered shielded microwave circuits," *Proc.IEEE Trans. Microwave Theory Tech.*, vol.54, no.1, pp.309–320, Jan. 2006.
- [30] V. Rizzoli, A. Costanzo, D. Masotti, A. Lipparini, and F. Matri, "Computer-aided optimization of nonlinear microwave circuits with the aid of electromagnetic simulation," *IEEE Trans. Microwave Theory Tech.*, vol.52, no.1, pp.362–377, Jan. 2004.
- [31] V. Rizzoli, A. Neri, D. Masotti, and A. Lipparini, "A new family of neural network-based bidirectional and dispersive behavioral models for nonlinear RF/microwave subsystems," *Int. J. RF Microwave Comput.-Aided Eng.*, vol.12, no.1, pp.51–70, Jan. 2002.
- [32] G. L. Creech, B. J. Paul, C. D. Lesniak, T. J. Jenkins, and M. C. Calcaterra, "Artificial neural networks for fast and accurate EM-CAD of microwave circuits," *IEEE Trans. Microwave Theory Tech.*, vol.45, no.5, pp.794–802, May. 1997.
- [33] Y. Wang, M. Yu, H. Kabir, and Q. J. Zhang, "Effective design of cross-coupled filter using neural networks ad coupling matrix," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp.1431–1434, Jun. 2006.
- [34] H. Kabir, Y. Wang, M. Yu, and Q. J. Zhang, "Neural network inverse modeling and applications to microwave filter design," *IEEE Trans. Microwave Theory Tech.*, vol.56, no.4, pp.867–879, Apr. 2008.
- [35] M. M. Vai, S. Wu, B. Li, and S. Prasad, "Reverse modeling of microwave circuits with bidirectional neural network models," *IEEE Trans. Microwave Theory Tech.*, vol.46, pp.1492–1494, Oct. 1998.
- [36] H. Ninomiya, "Microwave neural network models using improved online quasi-Newton training algorithm," *Journal of Signal Processing*, vol.15, no.6, pp.483–488, Nov. 2011.
- [37] H. Sharma and Q. J. Zhang, "Transient electromagnetic modeling using recurrent neural network," *IEEE MTT-SIMS Digest*, pp.1597–1600, Jun. 2005.
- [38] W. J. R. Hoefer and P. P. M. So, "The MEFiTo-2D Theory," Victoria, BC, Canada: Faustus Scientific Corporation, 2001.
- [39] T. Liu, S. Boumaiza, and F. M. Ghannouchi, "Dynamic behavioral modeling of 3G power amplifier using real-valued time delay neural networks," *IEEE Trans. Microwave Theory Tech.*, vol.52, no.3, pp.1025–1033, Mar. 2004.
- [40] M. Isaksson, D. Wisell, and D. Ronnow, "Wide-band modeling of power amplifiers using radial-basis function neural networks," *IEEE Trans. Microwave Theory Tech.*, vol. 53, no. 11, pp. 3422–3428, Nov. 2005.
- [41] B. O'Brien, J. Dooley, and T. J. Brazil, "RF power amplifier behavioral modeling using a globally recurrent neural network," *IEEE MTT-S Int. Microwave Symp. Dig.*, pp.1089–1092, Jun. 2006.
- [42] J. Wood, D. E. Root, and N. B. Tuffillaro, "A behavioral modeling approach to nonlinear model-order reduction for RF/microwave ICs and systems," *IEEE Trans. Microwave Theory Tech.*, vol.52, no.9, pp.2274–2284, Sep. 2004.
- [43] P. M. Watson and K. C. Gupta, "EM-ANN models for microstrip vias and interconnects in dataset circuits," *IEEE Trans. Microwave Theory Tech.*, vol.44, no.12, pp.2495–2503, Dec. 1996.
- [44] S. Roweis, "Levenberg-marquardt optimization," *Notes, University Of Toronto*, 1996.
- [45] M. K. Transtrum and J. P. Sethna. "Improvements to the Levenberg-Marquardt algorithm for nonlinear least-squares minimization," arXiv preprint arXiv:1201.5885, Jan. 2012.
- [46] M. I. A. Lourakis, "A brief description of the Levenberg-Marquardt algorithm implemented by levmar," *Foundation of Research and Technology*, pp.1–6, Feb. 2005.
- [47] D. H. Li and M. Fukushima, "A modified BFGS method and its global convergence in nonconvex minimization," *Journal of Computational and Applied Mathematics*, vol.129, pp.15–35, 2001.
- [48] D. Gao, N. Ruan, and W. Xing, editors, "Advances in Global Optimization," Springer Proceedings in Mathematics & Statistics, 2014.

- [49] *Sonnet*, Full-wave 3D Planar Electromagnetic Field Solver Software for High Frequency EM Simulation, Sonnet Software, Inc.

Analyzing Collaborative Learning Process by Deep Learning Methods: A Multi-Dimensional Coding Scheme with an Assessment Model

Taketoshi Inaba, Chihiro Shibata
Graduate School of Bionics, Computer and Media Sciences
Tokyo University of Technology
Tokyo, Japan
email: {inaba, shibatachh}@stf.teu.ac.jp

Kimihiko Ando
Cloud Service Center
Tokyo University of Technology
Tokyo, Japan
email: ando@stf.teu.ac.jp

Abstract—In computer-supported collaborative learning research, it may be a significantly important task to figure out guidelines for carrying out an appropriate scaffolding by extracting indicators for distinguishing groups with poor progress in collaborative process upon analyzing the mechanism of interactive activation. And for this collaborative process analysis, labelling for appropriately representing properties of each contribution (coding) and statistical analysis are often adopted as a method. But as far as this paper is concerned, it tries to automate this huge laborious coding work with deep learning technology. In its previous research, supervised data was prepared for deep learning based on a coding scheme consisting of 16 labels according to speech acts. In this paper, with a multi-dimensional coding scheme with five dimensions newly designed aiming at analyzing collaborative learning process more comprehensively and multilaterally, an automatic coding is performed by deep learning methods and its accuracy is verified. The results indicate with certainty that we can introduce this model to authentic educational settings and that even for large classes with many students, we can perform real-time monitoring of learning process or ex-post analysis of big educational data. However, presenting raw results of automatic coding on each dimension is not enough to indicate the collaborative process quality to teachers and students. Therefore, a new rating model that can assess and visualize the quality of collaborative process is proposed.

Keywords—CSCL; coding scheme; deep learning methods, automatic coding

I. INTRODUCTION

This article is an extended version of a conference paper presented at eLmL 2018, the Tenth International Conference on Mobile, Hybrid and On-line Learning [1]. It introduces more information on the related work of this study and especially a new proposal for visualization of results realized by our automatic coding method.

A. Analysis on Collaborative Process

One of the greatest research topics in the actual Computer Supported Collaborative Learning (CSCL) research is to

analyze its social and cognitive processes in detail in order to clarify what kinds of knowledge and meanings were shared within a group as well as how and by what arguments knowledge construction was performed. In addition, it is also required to develop CSCL system and tools with scaffolding function which may activate collaborative process by utilizing such knowledge.

However, because main data for the collaborative process analysis include contributions over chatting, images and voices on tools such as Skype, and various outputs prepared in the course of collaborative learning, it is totally inadequate to perform just quantitative analysis in order to analyze such data. Therefore, CSCL research changed direction more or less to qualitative research [2]-[5].

As these qualitative studies often result in in-depth case study, however, they have a downside that it is not easy at all to derive guidelines with generality, which are applicable also to other contexts. Therefore, studies have been conducted in recent years based on an approach of verbal analysis in which labeling for appropriately representing properties (hereinafter referred to as coding) is performed to each contribution in linguistic data of certain volume generated over the collaborative learning from perspectives of linguistics and collaborative learning activities [6]. On the other hand, an advantage of the approach is its capability of quantitative processing for significantly large scale data while keeping qualitative perspective. However, it is a task requiring significant time and labor to perform coding manually and it is expected to become impossible to perform coding manually in a case that data becomes further bigger in size.

In our research project, we have achieved certain results in a series of previous studies reported last year in eLmL 2017 and the like, using deep learning technique for automatic coding of vast amount of collaborative learning data [7][8][9]. In this paper, while verification is performed for accuracy of the automatic coding based on deep learning technique similarly to last year, supervised data has been constructed by conducting coding manually depending on adopted multi-dimensional coding scheme in order to newly analyze collaborative learning process in a more multilateral and comprehensive manner.

B. Purpose of This Study

The final goal of our research project is to implement support at authentic learning and educational settings such as real time monitoring of collaborative process and scaffolding for inactive groups based on analyses of large scale collaborative learning data as mentioned above.

As a further development of our previous research, a technique for automatizing coding of chat data is developed based on a multi-dimensional coding scheme capable of expressing collaborative learning process more comprehensively and its accuracy is verified in this study.

Specifically, after newly performing coding manually for substantial amount of the same chat data, which was used in the previous studies, a part of it is learned as training data by deep learning methods and then automatic coding is conducted for the test data. For accuracy verification, we try to verify the accuracy of automatic coding by calculating precision and recall of automatic coding of test data in each dimension. We also evaluate what type of misclassification occurred frequently in each dimension.

C. Structure of This Paper

This paper is structured as follows. In Section II, we present the related work. The outline and results of our previous work are shown in Section III. Our coding scheme newly developed this time is described in Section IV. Section V presents the dataset with the statistics of the new coding labels assigned by the human coders. Our experiments and results of the study are shown in subsequent Section VI. Section VII proposes an assessment model of the quality of collaborative processes and envisage a possible visualization of this model. Finally, in Section VIII, we present the conclusion and future work to complete the paper.

II. RELATED WORK

Deep neural networks [10] often has been applied in the field of natural language processing. Text classification is an important task in natural learning processing, for which various deep learning methods have been exploited extensively in recent studies. There are various modifications using convolutional neural networks (CNNs) that are applied for text classification [11][12][13]. In usual methods, texts are basically fed into CNNs with word-level embedding. Recent studies [14][15] show that character-level embedding is also promising method especially when datasets is sufficiently large. Using recurrent neural network (RNN) is another promising approach to achieve highly accurate results in text classification tasks. Long short-term memory units (LSTMs) [16] and gated recurrent units (GRUs) [17] are sophisticated architecture developed recently to overcome the drawbacks of RNNs. The language models used those RNNs can significantly outperform statistical language models, such as n-grams. RNNs are applied to text classification in various ways [17][18][19][20]. For instance, Yang et al. used a bidirectional GRU with attention modeling by setting two hierarchical layers that consist of the word and sentence encoders [21].

In the field of CSCL, some researchers have tried to apply text classification technology to chat logs. The most representative studies would be Rosé and her colleagues' works [22][23][24]. For example, they applied text classification technology to a relatively large CSCL corpus that had been coded by human coders using the coding scheme with multiple dimensions, developed by Weinberger and Fisher [23][25]. McLaren's Argonaut project took a similar approach: he used online discussions coded manually to train machine-learning classifiers in order to predict the appearance of these discussions characteristics in the new e-discussion [26]. However, it should be pointed out that all these prior studies rely on the machine learning techniques before deep learning studies emerge.

III. PREVIOUS WORK OF THIS STUDY

Outline of our previous work [7] is shown below.

A. Conversation Dataset

Dataset for the study conducted last year is based on conversations among students participating in online collaborative learning. This data set is obtained from chat function of CSCL system originally developed by the authors for lectures in the university [27]. By the way, we will add that this data is also used in this paper. Usage situation of CSCL as the source of the dataset is shown in Table I. Since students participated in multiple classes, number of participant students is less than the number obtained by multiplying number of groups and that of group members.

TABLE I. CONTRIBUTIONS DATA USED IN THIS STUDY

Number of Lectures	7 Lectures
Member of Groups	3-4 people
Learning Time	45-90 minutes
Number of Groups	202 groups
Number of Students	426 students
Dataset	11504 contributions

B. Coding Scheme

According to a manual for coding prepared by the authors, a label was assigned to each contribution of chat. Any of the 16 types of labels as shown in Table II was assigned. The ratio of each label is shown in Figure 1.

C. Automatic Coding Approach Based on Deep Learning

In the previous study, we adopted three types of Deep Neural Network (DNN) structures: 1) Convolutional Neural Networks (CNN), 2) Long-Short Term Memory (LSTM) and 3) Sequence to Sequence (Seq2Seq). Of the three models, Seq2Seq model is a deep neural network consisting of two LSTM units called encoder and decoder, and learning of classification problem and sentence generation is performed by entering pairs of strings of words to each part [28][29]. For example, the pair corresponds to a sentence in certain language and its translated sentence in case of translation

system as well as to question sentence and response sentence in case of question and answer system, respectively.

In addition, a model based on Support Vector Machine (SVM), which is a traditional machine learning approach is used as a baseline. Accuracy of each model is verified by comparing automatic coding concordance rate and Kappa coefficient. About technology and experiment results in detail for each classification model, please refer to existing literatures of the authors [7][8][9].

TABLE II. LIST OF LABELS

Label	Meaning of label	Contribution example
Agreement	Affirmative reply	I think that's good
Proposal	Conveying opinion, or yes/no question	How about five of us here make the submission?
Question	Other than yes/no question	What shall we do with the title?
Report	Reporting own status	I corrected the complicated one
Greeting	Greeting to other members	I'm looking forward to working with you
Reply	Other replies	It looks that way!
Outside comments	Contribution on matters other than assignment contents / Opinions on systems and such	My contribution is disappearing already; so fast! / A bug
Confirmation	Confirm the assignment and how to proceed	Would you like to submit it now?
Gratitude	Gratitude to other members	Thanks!
Request	Requesting somebody to do some task	Can either of you reply?
Correction	Correcting past contribution	Sorry, I meant children
Disagreement	Negative reply	I think 30 minute is too long
Complaint	Dissatisfactions towards assignments or systems	I must say the theme isn't great
Switchover	A contribution to change event being handled, such as moving on to the next assignment	Shall we give it a try?
Joke	Joke to other members	You should, like, learn it physically? :)
Noise	Contribution that does not make sense	?meet? day???

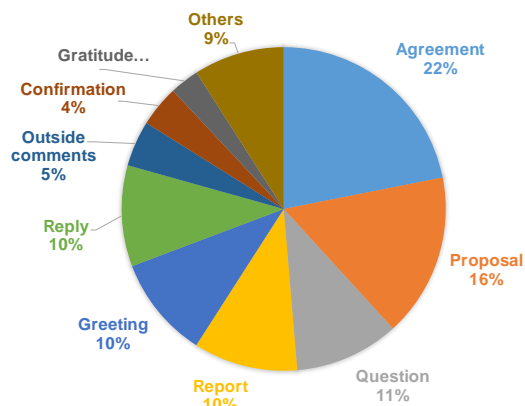


Figure 1. Ratio of each conversational coding labels

D. Experiment and Assessment

1) Outline of experiment

For the data set with manually prepared coding labels as described above, we compared the prediction accuracy of automatic coding for each model.

With separation of sentences into morpheme using MeCab conducted at first as a preprocessing of data, words with low use frequency were substituted by "unknown". Subsequently, just 8,015 contributions were extracted and 90% and 10% of them were sorted into data for training and test, respectively.

Naive Bayes, Linear SVM, and SVM based on RBF Kernel were applied as baseline approaches.

2) Experiment Results

Table III shows prediction accuracy (concordance rate) of models proposed in the previous study and those adopted as baseline for test data. The concordance rate here refers to a proportion that manually assigned label conforms with predicted label output by a model. It is proved, as Table III shows, that accuracy of the proposed model's result is higher than that of baseline model. Among the three models as described above, it is found that there is almost no difference in concordance rate between the approaches based on CNN with word vectors trained using the Wikipedia data slightly enhanced accuracy and LSTM (0.67-0.68). These approaches show concordance rates a little bit higher (around 2 to 3%) compared with SMV as a baseline approach (0.64-0.66).

On the other hand, a model based on Seq2Seq showed the highest concordance rate among all of the models (0.718), higher by 5 to 7% and 3 to 4% compared with SVM and other models, respectively.

TABLE III. PREDICTIVE ACCURACIES FOR BASELINES AND DEEP-NEURAL-NETWORK MODELS

Naive Bayes		SVM(Linear)		SVM(RBF Kernel)	
unigram	uni+bigram	unigram	uni+bigram	unigram	uni+bigram
0.554	0.598	0.642	0.659	0.664	0.659
CNN		LSTM		Seq2Seq	
with wikipedia	w.o. wikipedia	single-direction	bidirection	bidirection	bidir. w. intern.
0.686	0.677	0.676	0.678	0.718	0.717

Then, results as described above are discussed using Kappa coefficient, which is a measure of agreement between the two individuals (human and model in this case). At first, it may be said that LSTM model has achieved sufficiently higher result as the Kappa coefficient for the model shows 0.63. In general, Kappa coefficient of 0.8 or higher is believed to be preferable for utilizing automatic coding discrimination result by a machine in a reliable manner, however, further higher concordance rate is required. In case of Seq2Seq model, on the other hand, Kappa coefficient is 0.723 with great improvement, if not reaching 0.8.

The experiment results above have suggested that Seq2Seq model is superior to other approaches due to consideration for context information. Since Seq2Seq is a model with reply sources entered, it is believed that the improvement in the accuracy has been partly caused by not separate capturing of each contribution but consideration of the context information.

IV. NEW CODING SCHEME

As our previous studies mentioned some cases that Replay may include a meaning of Agree in the coding scheme, the fact that the definition of one label may sometimes overlap the definition of another label has become a factor making it difficult to assign a label always with accuracy and reliability. In addition to these technical problems, more importantly, labels based on speech acts, which express the linguistic characteristics of the conversation are insufficient for the analysis of the learning process. With this single linguistic scheme, it is almost impossible to realize whether members of

a group engage in activities to solve the task, how members coordinate each other in terms of task division, time management, etc. during their collaboration, how each member constructs his argument, how members discuss and negotiate each other. From those described above, we propose a new coding scheme so that the automated coding accuracy will improve and that we may understand more accurately and globally collaborative process.

Our new coding scheme is constructed based on the multi-dimensional coding scheme proposed by Weinberger et Fischer [25]. As shown in Table IV, our scheme consists of five dimensions, while Weinberger and Fischer's one has four dimensions without Coordination dimension. We provide labels basically regarding a contribution as a unit similarly to way we used in the previous studies. In addition, while such values as number of contributions are provided as Participation dimension labels, those in other four dimensions are provided by selecting one label from among multiple labels. In other words, since one label is given for each dimension for one contribution, a plurality of labels will be assigned to one contribution. Therefore, the coding work with this scheme is extremely complicated and takes a lot of time, but the merit of automated coding is even greater. Each dimension is described in detail below.

TABLE IV. NEW CODING SCHEME

Dimension	Description
Participation	Frequency of participation in argumentation
Epistemic	How to be directly involved in problem solving
Argumentation	Ideal assertion in argumentation
Social	How to cope with others' statements
Coordination	How to coordinate to advance discussion smoothly

A. Participation Dimension

Participation dimension is for measuring degree of participation in arguments. As this dimension is defined as quantitative data including mainly number of contributions and its letters, time of contributions, and interval of contributions, coding is performed by statistical processing on the database while requiring neither manual nor artificial intelligent coding. The list is shown in Table V.

TABLE V. PARTICIPATION DIMENSION

Category	Description
Number of contributions	Number of contributions of each member during sessions
Number of letters of a contribution	Number of letters during a single speech
Time for contribution	Time used for a contribution
Interval of contributions	Time elapsed since last contribution
contributions distribution	Standard deviation of each member within a group

Since Participation dimension labels handle number of specific contributions, it is possible to analyze quantitatively different aspects of participation in conversations but impossible to perform qualitative analysis such as whether the conversation contributed to problem solving.

B. Epistemic Dimension

This dimension shows whether each contribution is directly associated with problem solving as a task and the labels are classified depending on contents of the contributions as shown in Table VI. This dimension's labels are assigned to all contributions.

TABLE VI. LABELS IN EPISTEMIC DIMENSION

Label	Description
On Task	contributions directly related to problem solving
Off Task	contributions without any relationship with problem solving
No Sense	contributions with nonsensical contents

Weinberger and Fischer's scheme has 6 categories to code epistemic activities, which consist in applying the theoretical concepts to case information. But, as shown in Table VII, we set only two categories here, because we want to give generality by which we can handle as many problem-solving types as possible. "On Task" here refers to contributions directly related to problem solving and such contributions with contents as shown below belong to "Off Task".

- Contributions to ask meaning of problems and how to proceed with them
- Contributions to allocate different tasks to members
- Contributions regarding the system

Since Epistemic dimension represents whether directly related to problem solving, it works as the most basic code for qualitative analysis. In case of less "On Task" labels, for example, it is believed that almost no effort has been made for the task.

Besides, labels of Argument and Social dimensions are assigned when Epistemic dimension is "On Task", whereas those of Coordination dimension are assigned only when it is "Off Task". Coordination Dimension

Coordination dimension code is assigned only when Epistemic code is "Off Task" and it is also assigned to such contributions that relate to problem solving not directly but indirectly. A list of Coordination dimension labels is shown in Table VII but the labels are assigned not to all contributions of "Off Task" but just one label is assigned to such contributions that correspond to these labels. In addition, in case of replies to contributions with Coordination dimension labels assigned, labels of the same Coordination dimension are assigned.

"Task division" here refers to a contribution to decide who to work on which task requiring division of tasks for advancing problem solving. "Time management" is a contribution to coordinate degree of progress in problem solving, and for example, such contributions fall under the definition that "let's check it until 13 o'clock," and "how has it been in progress?" "Meta contribution" refers to a contribution for clarifying what the problem is when intention and meaning of the problem is not understood. "Technical coordination" refers to questions and opinions about how to use the CSCL System. "Proceedings" refer to contributions for coordinating the progress of the discussion.

Since Coordination dimension labels are assigned to such contributions that intend to problems smoothly, it is believed to be possible to predict progress in arguments by analyzing timing when the code was assigned. Further, in case of less labels of Coordination dimension, it may be predicted that smooth relationship has not been created within the group.

On the other hand, if a large number of these labels were assigned in many groups, it may be understood that there exists any defect in contents of the task or system.

TABLE VII. LABELS OF COORDINATION DIMENSION

Label	Description
Task division	Splitting work among members
Time management	Check of temporal and degree of progress
Technical coordination	How to use the system, etc.
Proceedings	Coordinating the progress of the discussion.

C. Argument Dimension

Labels of Argument dimension are provided to all contributions, indicating attributes such as whether each contribution includes the speaker's opinion and whether the opinion is based on any ground. Labels of this dimension are provided to just one contribution content without considering whether any ground was described in other contribution.

A list of Argument dimension labels is shown in Table VIII. Here, presence/absence of grounds is determined whether any ground to support the opinion is presented or not but it does not matter whether the presented ground is reliable or not. A qualified claim represents whether it is asserted that presented opinion is applied to all or part of situations to be worked on as a task. "Non-Argumentative moves" refer to contributions without including any opinion and simple questions are also included in this tag. Also, as a logical consequence, this label is assigned to all off-task contribution in the Epistemic dimension.

TABLE VIII. LABELS IN ARGUMENT DIMENSION

Label	Description
Simple Claim	Simple opinion without any ground
Qualified Claim	Opinion based on a limiting condition without any ground
Grounded Claim	Opinion based on grounds
Grounded and Qualified claim	Opinion with limitation based on grounds
Non-argumentative moves	contribution without containing opinion (including questions)

Labels in Argument dimension are capable of analyzing the logical consistency of contribution contents. For example, if a contribution is filled just with "Simple Claim" it is assumed as a superficial argument.

In comparison with Weinberger and Fischer's scheme, we do not set for now the categories of macro-level dimension in which single arguments are arranged in a line of argumentation such as arguments, counterarguments, reply, for the reason that it seems difficult that the automatic coding by deep learning methods for this macro dimension works correctly. Social Dimension

Labels in Social dimension are provided when Epistemic code is "On task" but they are provided not to all contributions "On task" but to a contribution which conforms to Epistemic code. This dimension represents how each contribution is related to those of other members within the group. Therefore, it is required to understand not only a contribution but also the previous context. Table IX shows a list of labels of the dimension.

"Externalization" refers to contributions without reference to other's contributions and it is assigned to contributions to be an origin of arguments mainly at the start of argument on a topic. "Elicitation" is assigned to such contributions that request others for extracting information including question. "Consensus building" refers to contributions that express certain opinion in response to other's contribution and they are classified into the three labels below. "Quick consensus building" is assigned to such contributions that aim to form prompt consensus with other's opinion. It is assigned to a case to give consent without any specific opinion. "Integration-oriented consensus building" is assigned to such contributions that intend to form consensus with other's opinion while adding one's own opinion. "Conflict-oriented consensus building" is assigned to such contributions that confront with other's opinion or request revision of the opinion. "Summary" is assigned to contributions that list or quote contributions that have been posted.

Since Social dimension code represents involvement with others, it may be understood how actively the argument was developed or whose opinion within the group was respected by analyzing Social dimension labels. For example, it may be assumed that arguments with frequent "Quick consensus building" result in accepting all opinions provided with almost no deep discussion.

TABLE IX. CODE OF SOCIAL DIMENSION

Label	Description
Externalization	No reference to other's opinion
Elicitation	Questioning the learning partner or provoking a reaction from the learning partner
Quick consensus building	Prompt consensus formation
Integration-oriented consensus building	Consensus formation in an integrated manner
Conflict-oriented consensus building	Consensus forming based on a confrontational stance
Summary	Statment listing or quoting contributions

D. Learning for each code granting and artificial intelligence

In the new coding scheme, "Participation" dimension labels are automatically generated from contribution logs, whereas other labels require manual coding by human coders in order to build up training data for deep learning and test data. Further, labels to be provided are decided by selecting from any of the dimensions of "Argument", "Social" and "Coordination" depending on a result of "Epistemic" labels. "Argument" and "Social" dimension labels are provided if the "Epistemic labels are "On task." In a case that "Epistemic" labels are "Off task", those in "Coordination" dimension are provided.

V. DATASET AND STATISTICS

A. Target Dataset

The raw dataset is taken from the real conversation log of the CSCL system, which is the same one as that of previous study (Table I). On this dataset, the coding labels were newly annotated based on the new coding scheme. Labeling was manually carried out by several people in parallel. The human coders were lectured about the new coding scheme by a professional in advance in order to code labels as accurately as possible. To evaluate the accuracy of the manual coding, we had each contribution annotated by two annotators and measured the coincidence rate for each dimension of the new coding scheme.

B. Manual Coding and Preprocessing

While 9,962 contributions were manually coded in all, some contributions do not make sense as a text of CLSL. For instance, the duplicated posts, the blank posts, and the contributions that consist of only ASCII art can be mentioned. Such kinds of contributions were marked as "non-sense" when the annotators labeled, and removed or simply ignored when the computer read them. After that, 9,197 contributions were remained as the useful data, on which the substantial jobs such as learning and classification are feasible.

The coincidence rates of the coding labels given by two human coders are significant for understanding the difficulty of the prediction, as well as to see the correctness of the manually coded labels. Table X shows the coincidence rate, the number of the valid contributions, and that of the coincidence contributions for each dimension. For the Epistemic dimension, since the coincidence rate is high for human coders, we can expect that it is also easy for machines to classify them. On the other hand, for the Social dimension, since the coincidence rate is low for human coders and the valid samples are sparse, the opposite result is expected.

TABLE X. THE VALID CONTRIBUTIONS AND THE COINCIDENT CONTRIBUTIONS

	# of valid contributions	# of coincidence contributions	Rate
Epistemic	9,197	8,460	0.92
Argumentation	9,083	7,765	0.85
Coordination	4,543	3,510	0.77
Social	3,917	2,619	0.67

C. Statistics of the New Coding Labels

In this subsection, we describe the statistics of the new coding labels assigned by the human coders with respect to each dimension. As we have multiple coders classify them, the statistics depend on the coders. When making a dataset for machines, we limit the contributions so as to have the same label assigned by the human coders. Thus, we describe the statistics of such contributions.

The ratios of "On task" and "Off task" in the Epistemic dimension are shown in Figure 2. In our dataset, the 'On task' contributions were a bit fewer than the 'Off task.' This implies that, at least from the view point of the conversation

log, the cost of the communication was more than the cost of discussion in group work. Although this result is just an instance obtained by applying our CSCL system to the actual group works for limited lectures, we can at least conclude that the communication cost is not small in a group work.

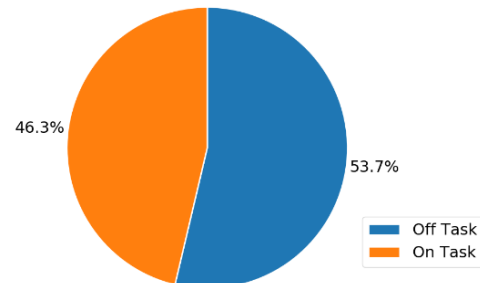


Figure 2. Ratio in the Epistemic dimension

Figure 3 shows the ratios of the labels in the Social dimension. Recall that its domain is On-task contributions. The label "Externalization" accounted half of the On-task contributions. The "Quick consensus building" followed it. Meanwhile, the ratios of the "Summary" and the "Consensus Buildings" except for the "Quick" one were small. These statistics show that the actual discussion mainly consisted of expressions of their opinions. Although we found that the contributions building consensus rarely come up in a real group work, we believe that they are the important keys for the discussion. Thus, we may can weight them when we assess the contribution to the discussion by students.

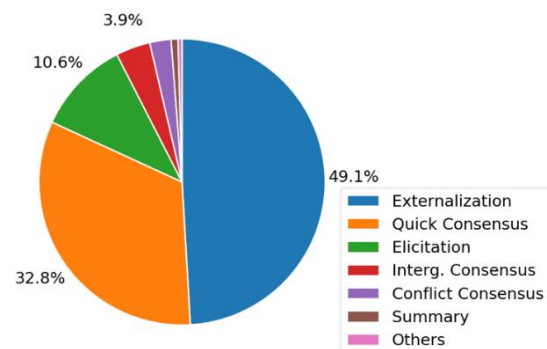


Figure 3. Ratio in the Social dimension

With respect to the "Coordination" dimension, the domain of which is the Off-task contributions, the most of them are assigned to "Other" as Figure 4 shows. The contributions labeled "Other" consist of short sentences that are not significant for neither discussion nor coordination of the group work. The representative examples are greetings and kidding. Meanwhile, the statistics show that the contributions except for "Other" also occupies more than a quarter. Since these kinds of contributions are related to coordinating tasks in the group work, they can be thought as important contributions for the assessment.

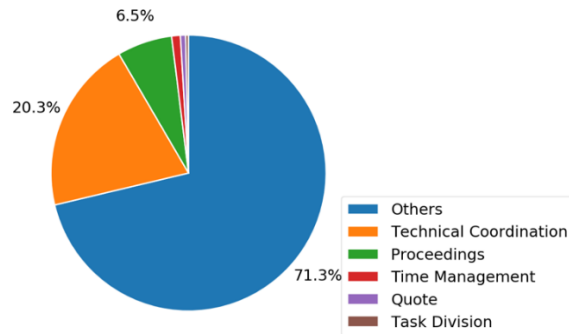


Figure 4. Ratio in the Coordination dimension

The labels in the "Argument" dimension are assigned independently of other dimensions. Thus, its domain spans both the On-task and the Off-task contributions. As shown in Figure 5, the label "Non-Argumentative moves" occupied more than 60 % of all. The label "Simple Claim" occupied the second percentage. To assess the discussion of the group work, at least it is necessary to remove the "Non-Argumentative" contributions and pay attention to which kind of claim is presented, even if almost every claim can be classified into the "Simple Claim". Therefore, the automatic coding for this dimension is as valuable as for the other three dimensions.

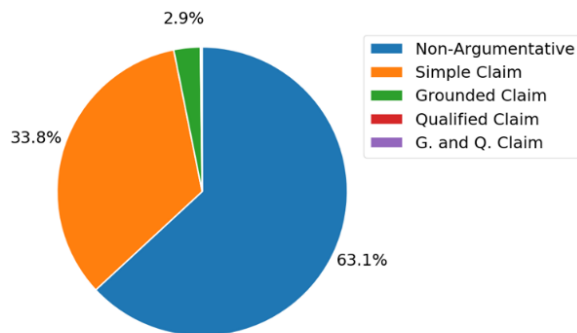


Figure 5. Ratio in the Argument dimension

VI. EXPERIMENTS

A. Approach to Learning and Classification

As described in Section II, deep neural networks (DNNs) outperform other machine learning methods significantly at least for the coding labels proposed by our previous studies [6][7][8]. Their results of the experiments show that the Seq2Seq-based model achieves the highest accuracy among several DNN structures. Thus, we apply the Seq2Seq-based model to classify our new coding labels in this paper.

The new coding scheme has four axes to be labeled as discussed in Section III; the Epistemic, the Coordination, the Argument, and the Social dimension. In the following experiments, the labels in each axis, or the dimensions, are learned and classified. There are solid dependencies among the Epistemic, the Coordination and the Social dimensions, while the Argument dimension is independent of the other dimensions. As shown in Figure 6, there is a dependency tree

among the former three dimensions. For instance, the label of the Social dimension is assigned only if that of the Epistemic is "On task." Therefore, the number of available contributions for learning is different for each classification task. In the following experiments, since we use the samples that have the coincidence labels only, the number of the available contribution was 8,460 for the Epistemic, 7,795 for the Augmentation, 3,510 for the Coordination, and 2,619 for the Social.

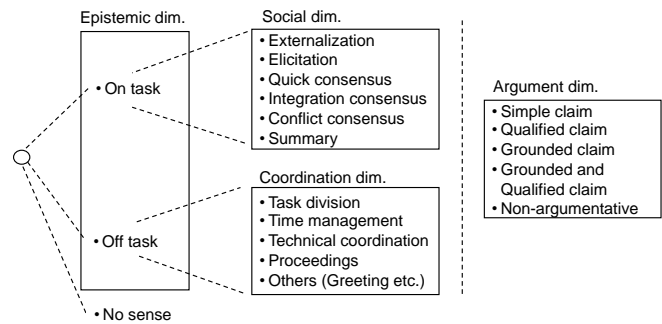


Figure 6. Dependency of Dimensions

B. Parameter Settings

We set the parameters for learning to the same values as in our previous study. They include the various kinds of the parameters such as the number of layers, the vector sizes of layers, the option of the optimization algorithms, learning rate, etc. The details can be referred to our previous studies [6][7][8].

C. Results for the Epistemic Dimension

The results of the experiments show that the On and Off tasks can be classified correctly with sufficiently high accuracy (Figure 7). The Seq2Seq-based model achieves more than 90 % in both precision and recall (Table XI). Since the coincidence ratio by two human coders is 91%, we can say that the accuracy of automatic coding, which is comparable to human beings was obtained for the Epistemic dimension.

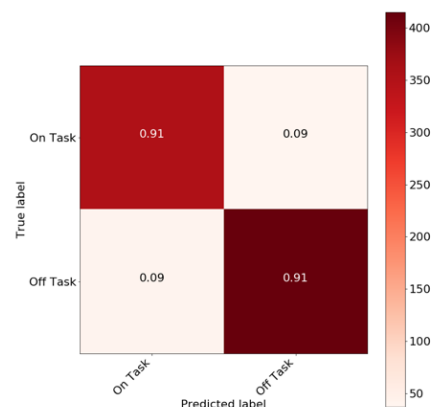


Figure 7. Confusion matrix for the Epistemic dimension

TABLE XI. PRECISION AND RECALL FOR THE EPSTEMIC DIMENTION

	Precision	Recall	F1-Score	Support
On Task	0.90	0.91	0.90	390
Off Task	0.92	0.91	0.91	456
Average(Micro) / Total	0.91	0.91	0.91	846

D. Results for the Argument Dimension

The classification accuracy is also high for the Argument dimension. The micro-averaged F1 score is 87 % (Table XII). Especially, the F1 score for the label "Non-argumentative Moves" is high sufficiently (92 %), which means that our model can surely recognize whether the contribution has any substantial meaning as a claim or not. On the other hand, while the precision for the "Simple Claim" is high (89 %), the recall for it is low (72 %). According to the confusion matrix shown in Figure 8, a quarter of the Simple Claim is misclassified into the Non-argumentative. This is because it is difficult to distinguish contributions that have a very small opinion from that have no opinions.

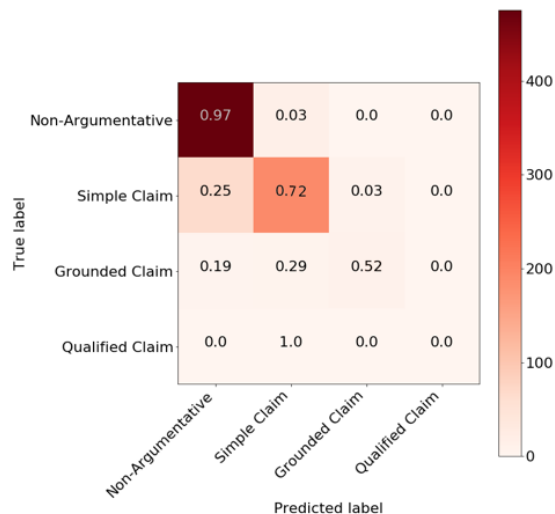


Figure 8. Confusion matrix for the Argument dimension

TABLE XII. PRECISION AND RECALL FOR THE ARGUMENT DIMENTION

	Precision	Recall	F1-Score	Support
Non-Argumentative	0.87	0.97	0.92	491
Simple Claim	0.89	0.72	0.80	264
Grounded Claim	0.58	0.52	0.55	21
Qualified Claim	0.00	0.00	0.00	1
Average(Micro) / Total	0.87	0.87	0.87	777

E. Results for the Coordination Dimension

Regarding the Coordination dimension, our model also achieved high classification accuracy. Seeing that the number of supports varies greatly among the labels, we should evaluate the classification ability of the model by the micro-averaged accuracies over all coding labels. As Table XIII shows, the micro-averaged F1 score was 85 %.

According to the results for each label (Figure 9), the following is observed. The major labels such as "Other" and

"Technical coordination" are classified correctly with high precisions, while the minor labels such as "Time Management", "Quote" and "Task Division" are not. Because the data for those minor labels are very limited, which have less than 50 contributions, it is quite difficult to learn them accurately. One of our future issues is to find some way to deal with those sparse labels.

TABLE XIII. PRECISION AND RECALL FOR THE COORDINATION DIMENTION

	Precision	Recall	F1-Score	Support
Others	0.91	0.91	0.91	242
Technical Coordination	0.81	0.80	0.81	82
Proceedings	0.58	0.70	0.64	20
Time Management	0.33	0.25	0.29	4
Quote	0.00	0.00	0.00	1
Task Division	0.00	0.00	0.00	2
Average(Micro) / Total	0.85	0.86	0.85	351

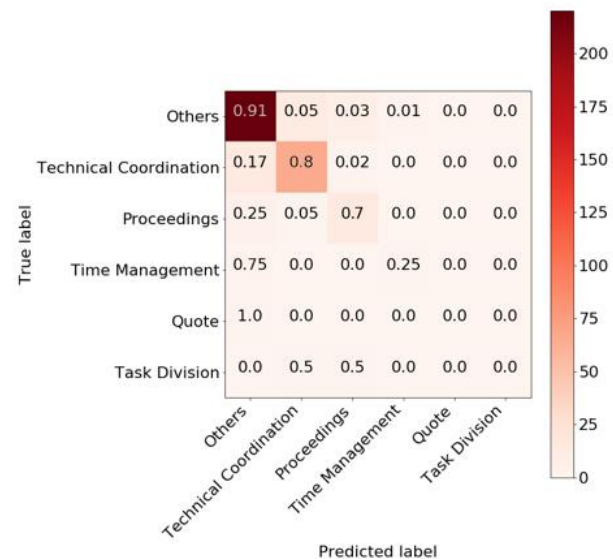


Figure 9. Confusion matrix for the Coordination dimension

F. Results for the Social Dimension

Comparing to the other dimensions, the accuracy was relatively low for the Social dimension. The F1 score was 70 % (Table XIV). Since labeling the Social sometimes needs understanding the deep meaning of the contribution and the background story of the discussion, it seems to be difficult for machines to learn them correctly with limited data.

According to Figure 10, the recall of the label "Externalization" is especially low (61 %), while those of "Quick Consensus" and "Elicitation" are high sufficiently (93 % and 97 %, respectively). According to the confusion matrix in Figure 10, there is a major reason that worsen the accuracy; the Externalization labels are easily misclassified to the Quick Consensus and to the Elicitation, but not vice versa. This fact also explains the reason why the precisions for the Quick Consensus and the Elicitation are low though the recalls for them are high. To improve the result, it is necessary to pursue the causes of these two types.

TABLE XIV. PRECISION AND RECALL FOR THE SOCIAL DIMENSION

	Precision	Recall	F1-Score	Support
Externalization	0.86	0.61	0.72	127
Quick	0.71	0.93	0.81	88
Elicitation	0.56	0.97	0.71	29
Interg. Consensus	0.17	0.14	0.15	7
Conflict Consensus	0.00	0.00	0.00	6
Summary	0.00	0.00	0.00	3
Others	0.00	0.00	0.00	2
Average(Micro) / Total	0.72	0.72	0.70	262

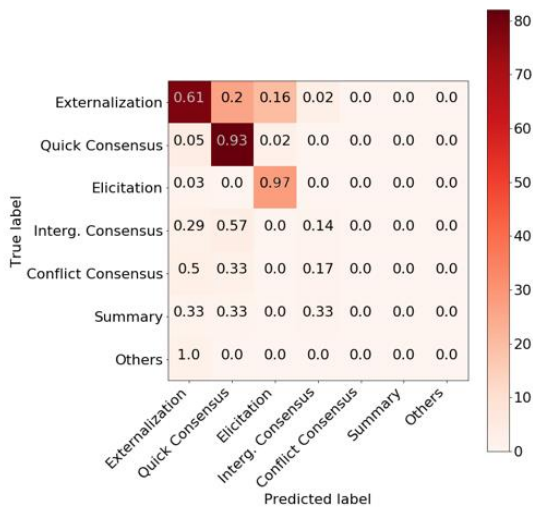


Figure 10. Confusion matrix for the Social dimension

VII. ASSESSMENT MODEL

The method of automating multi-dimensional coding proposed in this study shows the technical possibility to clarify the situation of collaborative learning in real time from various perspectives even if it targets big educational data. However, as mentioned in the introduction, one of the ultimate goals of this research is to automatically analyze the collaborative learning process in real time and show the results to teachers and learners in an easy-to-understand manner. To that end, it is not enough to merely automate and speed up coding, and these coding data needs to be reintegrated and visualized in some form into a "readable" format.

So, we refer to the rating scheme for collaborative process assessment proposed by Meir, Spada, Rummel and we propose a model that adapts this scheme to the context of our research [30][31]. There are two reasons for choosing this scheme. First, in the empirical assessment, it is shown that positive findings exist for inter-rater reliability, consistency, and validity of this scheme. Second, as these authors have already recommended, this rating scheme is designed assuming that it will be customized according to various collaborative learning situations.

When designing the rating scheme, they define five aspects as factors of successful collaborative learning from the content analysis of empirical data and theoretical consideration based on the survey of the learning theory

literature. That is, Communication, Joint information processing, Coordination, Interpersonal relationship, Motivation. In addition, as shown in Table XV, nine assessment dimensions are set for these five aspects. In these assessment dimensions, quantitative assessment is performed on a five point grade scale respectively.

TABLE XV. FIVE ASPECTS OF THE COLLABORATIVE PROCESS AND THE RESULTING NINE DIMENSIONS OF MEIR, SPADA AND RUMMEL'S RATING SCHEME

Process dimensions
A.Communication
1) Sustaining mutual understanding
2) Dialogue management
B. Joint information processing
1) Information pooling
2) Reaching consensus
C.Coordination
1) Task division
2) Time management
3) Technical coordination
D. Intrepersonal interaction
1) Reciprocal interaction
E. Motivation
1)Individual task orientation

Since this paper is not a place to discuss the details of this original rating scheme, we will briefly describe the aspects and dimensions proposed in our research below. Regarding the aspects, it follows the original scheme. About the definition of "assessment targets" (we use this term to avoid confusion with dimensions of coding scheme), considering the fact that the major fields of our research are lectures at large university classrooms and the fact that there is no significant difference between students in knowledge level before class, some customizations are done. Also, in each assessment target, which coding data is referred to is also described. All of the nine assessment targets shown below are assumed to be quantified on a five-point grade scale, and the eight targets other than the last Individual task orientation can be easily visualized with an octagonal radar chart as shown in Fig.11.

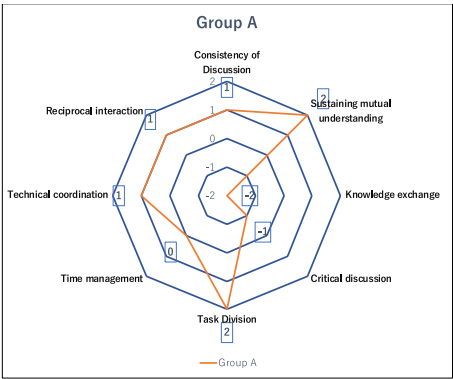


Figure 11. Example of visualization of collaborative process assessment

A. Communication

In order to facilitate the discussion within the group, it is necessary that the basic concepts of the task and the hypotheses and problems that have appeared in discussions are shared. To that end, it is very important to repeatedly confirm that discussions are progressing on a common ground. Especially in conversation in chat, unlike face to face, members have to confirm mutual understanding more explicitly and frequently. We propose the following two as assessment targets of this smooth and "grounded" conversation.

1) Consistency of Discussion

For consistency of discussion, we evaluate whether discussions between members progress in a smooth flow. In a smoothly advancing discussion, other members should pay attention to others' contributions and return replay.

The reaction to the contribution by others is mainly expressed by the labels of Social dimension. To evaluate this target, it is good to refer to the following data and quantify it:

- Number of contributions in Social dimension and their frequency in On Task statements in Epistemic dimension;
- Number of contributions belonging to Social Dimension immediately after Social dimension's label "Externalization" and their frequency.

2) Sustaining mutual understanding

In maintaining mutual understanding, it is required to confirm in the process of discussion whether the understanding about basic concepts and problem awareness are shared between the members. Therefore, it is necessary to question each other, obtain answers, and confirm mutual understanding. Also, if there is a misunderstanding, it is necessary to resolve this.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions labeled as Elicitation in Social Dimension and frequency of contributions belonging to Social Dimension immediately after the Elicitation contribution.

However, the assessment of this target is insufficient in the current coding scheme and we will need to set a new coding label.

B. Joint Information Processing

In problem solving for collaborative learning, it is required that each member provides his own knowledge, or obtains knowledge that he does not have from others to reach a more advanced solution. Therefore, the learner needs to explain knowledge in such a way that other members can understand. Also, in the process of problem solving, it is necessary to make questions and counterarguments to each other's opinions, and also to consider alternative solutions to make final decision making. The following two assessment targets are proposed as assessment targets of this aspect.

1) Knowledge Exchange

In this target, we assess to what extent members provided each other's knowledge in such a way that other members can understand their own knowledge.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions labeled as Grounded, Qualified, Grounded and Qualified in the Argument dimension and their frequency.

2) Critical Discussion

This target assesses if easy compromise is avoided and to what extent counterarguments and integrations to each other's views are effectuated.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions labeled as Conflict-oriented consensus building and Integration-oriented consensus building in the Social dimension and their frequency.

C. Coordination

In order to succeed in collaborative learning, it is extremely important to plan ahead in advance of the whole work, to share subtasks with members, and to effectively manage time. In CSCL, not only chat but also interfaces such as file sharing and common workspace are generally prepared, so that technical coordination of these work environments also needs to be done successfully. As the assessment targets of this aspects, the following three assessment targets are adopted. In these three assessment targets, the small number of contributions is an indicator that coordination is insufficient. But if the number of contributions is too numerous, the problem solving activities (On task) may be not sufficiently carried out. Therefore, there is room for discussion about determining the appropriate number of contributions and their frequency.

1) Task Division

In this target, it is evaluated whether work division between members is done well.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions labeled as Task division in the Coordination dimension and their frequency.

2) Time Management

In this target, we evaluate how the members manage the time constraints and work.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions and frequency of Time management label in the Coordination dimension.

3) Technical Coordination

In this target, we evaluate whether effective use of technical resources is realized.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions labelled as Technical coordination in the Coordination dimension and their frequency.

D. Interpersonal Relationship

In order for collaborative learning to succeed, it is necessary to exchange frankly opinions among participants; even if opinions are conflicting, it should be avoided that human relationship itself between members becomes confrontational. Participants must respect each other and behave in a friendly manner.

For the assessment of this aspect, the following assessment target is adopted. But it seems that it is insufficient to accurately capture this aspect in the current coding scheme. However, it is easy to automatically identify contributions such as greetings or thanks.

1) Reciprocal Interaction

In this target, we evaluate whether members are speaking equally or whether each participates evenly in problem solving and decision making.

For the evaluation of this target, we will refer to the following data and quantify it:

- Number of contribution and contribution distribution in Participation dimension.

E. Motivation

In order to animate the group as a whole, it is necessary for each member to act positively on problem solving and encourage other members to actively participate. However, there are significant differences between members such as efforts to solve problems, participation in discussions, and encouragement to other group members. In order to evaluate the aspect of this individual contribution, the following assessment target is adopted.

1) Individual Task Orientation

In this target, the contribution degree of each group member is individually assessed.

For the assessment of this target, we will refer to the following data and quantify it:

- Number of contributions labelled as On task in the Epistemic dimension of each member.

VIII. CONCLUSION AND FUTURE WORK

A. Conclusion

In this study, we proposed a newly designed coding scheme with which we tried to automate time-consuming coding task by using deep learning technology.

We have constructed a new coding scheme with five dimensions to analyze different aspects of the collaboration process. After manually coding a large volume dataset, we proceeded to the machine learning of this dataset using Seq2seq model. Then, we evaluated the accuracy of this automatic coding in each dimension. Except some typical types of the misclassifications, the results were overall positive. If this misclassification is resolved to a considerable extent, it will also come into view to apply this technique in real educational settings and for large classes with many students in order to perform real-time monitoring of learning process or ex-post analysis of big educational data.

Finally, at the end of the paper, we propose a new assessment model that can assess and visualize the quality of collaborative process.

B. Future Work

As for the future research directions, we may have three areas to pursue.

The first area is about some typical misclassifications in the Social Dimension. To improve prediction accuracy, one could make more explicit and comprehensible the referential relation between a contribution and others even for the machines, if one indicates contributions to which a contribution refers. For example, with regard to the typical misclassification mentioned above between "Externalization" and "Quick Consensus" or "Elicitation", since contributions labeled "Externalization" have no reference to other contributions, we can hope to effectively reduce these misclassifications with this kind of indicator. In addition, as the next step of this paper, it seems to be worth trying to compare the accuracy using DNN models other than Seq2seq and other network structures such as memory networks [32].

The second area concerns the intrinsic structure of our coding scheme. Since the scheme contains different dimensions and under each dimension different labels are hierarchically organized, it is very interesting to discover not only correlations among dimensions, but also among labels belonging to different dimensions [33]. If we can input the information about the correlation between such labels in some form at the time of automatic classification, the accuracy of automatic coding can be further improved.

The third area relates to the assessment method and its visualization of collaborative process. In this paper, the method of calculating the rating of assessment targets is not defined yet, which is an urgent task. Furthermore, we will have to reconsider which data should be referenced for each target. Also, it may be necessary to partially modify the scheme itself to fit the assessment model. For visualization, we should consider not only visualization of real-time collaborative situation but also design method to intuitively visualize transition on time axis and comparison between different groups.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 26350289, 17H02004 and 16K01134.

REFERENCES

- [1] T.Kanayama, Ch.Shibata, K.Ando, and T.Inaba, "Using deep learning methods to automate collaborative learning process coding based on multi- dimensional coding scheme," The Tenth International Conference on Mobile, Hybrid, and On-line Learning, pp. 45-53, 2018.
- [2] G. Stahl, T. Koschmann, and D. Suthers, "Computer-supported collaborative learning," In The Cambridge handbook of the learning science, K. Sawyer, Eds. Cambridge university press, pp. 479-500, 2014.

- [3] P. Dillenbourg, P. Baker, A. Blaye, and C. O'Malley, "The evolution of research on collaborative learning," In *Learning in humans and machines: Towards an interdisciplinary learning science*, P. Reimann and H. Spada, Eds. Oxford: Elsevier, pp. 189-211, 1996.
- [4] T. Koschmann, "Understanding understanding in action," *Journal of Pragmatics*, 43, pp. 435-437, 2011.
- [5] T. Koschmann, G. Stahl, and A. Zemel, "The video analyst's manifesto (or The implications of Garfinkel's policies for the development of a program of video analysis research within the learning science)," In *Video research in the learning sciences*, R. Goldman, R. Pea, B. Barron and S. Derry, Eds. Routledge, pp. 133-144, 2007.
- [6] M. Chi, "Quantifying qualitative analyses of verbal data : A practical guide ," *Journal of the Learning Science*, 6(3), pp. 271-315, 1997.
- [7] C. Shibata, K. Ando, and T. Inaba, "Towards automatic coding of collaborative learning data with deep learning technology", *The Ninth International Conference on Mobile, Hybrid, and On-line Learning*, 2017, pp. 65-71.
- [8] K. Ando, C. Shibata, and T. Inaba, "Analysis of collaborative learning processes by automatic coding using deep learning technology", *Computer & Education*, 43, pp. 79-84, 2017.
- [9] K. Ando, C. Shibata, and T. Inaba, "Coding collaborative learning data automatically with deep learning methods", *JSi SE Research Report*, 32, 2017.
- [10] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, 521(7553), pp. 436-444, 2015.
- [11] Y. Kim, "Convolutional neural networks for sentence classification," *arXiv preprint arXiv:1408.5882*, 2014.
- [12] P. Zhou, Z. Qi, S. Zheng, J. Xu, H. Bao, and B. Xu, "Text classification improved by integrating bidirectional lstm with two-dimensional max pooling". In *Proceedings of COLING 2016*, 2016.
- [13] R. Johnson and T. Zhang, "Deep pyramid convolutional neural networks for text categorization", In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pp. 562-570 2017.
- [14] X. Zhang, J. Zhao, and Y. LeCun. "Character-level convolutional networks for text classification," In *Proceedings of the 28th International Conference on Neural Information Processing Systems (NIPS2015)*, pp. 649-657, 2015.
- [15] A. Conneau, H. Schwenk, Y. LeCun and L. Barrout, "Very Deep Convolutional Networks for Text Classification," In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pp. 1107-1111, 2017
- [16] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, 9(8), pp. 1735-1780, 1997.
- [17] J. Chung, C. Gulcehre, K. Hyun Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *arXiv preprint arXiv:1412.3555*, 2014.
- [18] Z. Yang et al., "Hierarchical Attention Networks for Document Classification," In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics (NAACL2016)*, Human Language Technologies, 2016.
- [19] D. Tang, B. Qin, and T. Liu, "Document modeling with gated recurrent neural network for sentiment classification," In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP2016)*, pp. 1422-1432, 2015.
- [20] R. Johnson and T. Zhang, Supervised and semi-supervised text categorization using lstm for region embeddings. In *International Conference on Machine Learning*, pp. 526-534, 2016.
- [21] J. Howard and S. Ruder, "Universal Language Model Fine-tuning for Text Classification", In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pp. 328-339, 2018.
- [22] C. Rosé et al., "Towards an interactive assessment framework for engineering design project based learning," In *Proceedings of DETC2007*, 2007.
- [23] C. Rosé et al., "Analyzing collaborative learning processes automatically: Exploiting the advances of computational linguistics in computer-supported collaborative learning," *International Journal of Computer Supported Collaborative Learning*, 3(3), pp. 237-271, 2008.
- [24] G. Gweon, S. Soojin, J. Lee, S. Finger and C. Rosé, "A framework for assessment of student project groups on-line and off-line," In *Analyzing Interactions in CSCL: Methods, Approaches and Issues*, S. Putambekar, G. Erkens and C. Hmelo-Silver Eds. Springer, pp. 293-317, 2011.
- [25] A. Weinberger and F. Fischer, "A frame work to analyze argumentative knowledge construction in computer-supported learning," *Computer & Education*, 46(1), pp. 71-95, 2006.
- [26] B. McLaren, O. Scheuer, M. De Laat, H. Hever and R. De Groot, "Using machine learning techniques to analyze and support mediation of student e-discussions," In *Proceedings of artificial intelligence in education*, 2007.
- [27] T. Inaba and K. Ando, "Development and evaluation of CSCL system for large classrooms using question-posing script," *International Journal on Advances in Software*, 7(3&4), pp. 590-600, 2014.
- [28] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv*, pp. 1409.0473, 2014.
- [29] O. Vinyals and Q. V. Le, "A Neural Conversational Model," *arXiv preprint arXiv:1506.05869*, (ICML Deep Learning Workshop 2015), 2015.
- [30] A. Meir, H. Spada, and N. Rummel, "A rating scheme for assessing the quality of computer-supported collaboration processes," *International Journal of Computer-Supported Collaborative Learning*, 2(1), pp. 63-86, 2007.
- [31] N. Rummel, A. Deiglmayr, H. Spada, G. Kahrmanis, and N. Avouris, "Analyzing collaborative interactions across domains and settings: an adaptable rating scheme," in *Analyzing interactions in CSCL*, S. Putambekar et al, Eds. Springer, pp. 367-390, 2011.
- [32] S. Sukhbaatar, A. Szlam, J. Weston and R. Fergus, "End-to-end Memory Networks," *Proceedings of the 28th International Conference on Neural Information Processing Systems*, pp. 2440-2448, 2015.
- [33] F. Scafino, G. Pio, M. Ceci, and D. Moro, "Hierarchical multi-dimensional classification of web documents with MultiWebClass," *International Conference on Discovery Science*, pp. 236-250, 2015.

Visibility Velocity Obstacles (VVO): Visibility-Based Path Planning in 3D Environments

Oren Gal, Yerach Doytscher

Mapping and Geo-information Engineering
Technion - Israel Institute of Technology
Haifa, Israel
e-mail: {orengal,doytscher}@technion.ac.il

Abstract - In this paper, we present as far as we know for the first time, a unique method combining visibility analysis in 3D environments with dynamic motion planning algorithm, named Visibility Velocity Obstacles (VVO). Our method is based on two major steps. The first step is based on analytic visibility boundaries calculation in 3D environments, taking into account sensors' capabilities including probabilistic consideration. In the second stage, we generate VVO transferring visibility boundaries from the position space to the velocity space, for each object. Each VVO represents velocity's set of possible future collision and visibility boundaries. Based on our analysis in velocity space, we plan our trajectory by selecting future robot's velocity at each time step, tracking after specific target considering visibility constraints as integral part of the velocities space. We formulate the tracked target in the environment as part of our planner and include visibility analysis for the next time step as part of our planning in the same search space. For the first time, we define visibility aspects as part of velocity space, where all the objects are modeled from visibility point of view. We introduce potential trajectory planner combining unified 3D visibility analysis for target tracking as part of dynamic motion planning.

Keywords - *Visibility; Motion planning; 3D; Urban environment; Spatial analysis.*

I. INTRODUCTION

Trajectory planning has developed alongside the increasing numbers of Unmanned Aerial Vehicles (UAVs), drones unmanned ground vehicles all over the world, with a wide range of applications such as surveillance, information gathering, suppression of enemy defenses, air to air combat, mapping buildings and facilities, etc.

Most of these applications are involved in very complicated environments (e.g., urban), with complex terrain for civil and military domains [1].

With these growing needs, several basic capabilities must be achieved. One of these capabilities is the need to avoid obstacles such as buildings or other moving objects, while autonomously navigating in 3D urban environments.

Path planning problems have been extensively studied in the robotics community, finding a collision-free path in static or dynamic environments, i.e., moving or static obstacles.

Over the past twenty years, many methods have been proposed, such as starting roadmap, cell decomposition, and potential field [6].

In this paper, we present visibility aspects as part of velocity space, where all the objects are modeled from visibility point of view. We introduce potential trajectory planner combining unified 3D visibility analysis for target tracking as part of dynamic motion planning. In the first part, we formulate visibility boundaries problem and introduce analytic solution that in the following sub-section integrated with sensor's limitations. Later on, we present the VVO method, demonstrated with visibility boundaries with cars, pedestrians and buildings visibility boundaries. In the last part, we suggest pursuer planner using VVO for UAV test case.

II. RELATED WORK

Path planning becomes trajectory planning when a time dimension is added for dynamic obstacles [7][8]. Later on, a vehicle's dynamic and kinematic constraints have been taken into account, in a process called kinodynamic planning [9]. All of these methods focus solely on obstacle avoidance.

Trajectory planning for air traffic control and ground vehicles has been well studied [10], based on short path algorithms using 2D polygons, 3D surfaces [11]. UAVs navigation has also been explored with vision-based methods [12], with local planning or a predefined global path [13].

UAV path planning is different from simple robot path planning, due to the fact that a UAV cannot stop, and must maintain its velocity above the minimum, as well as not being able to make sharp turns.

UAV path planning methods usually decompose the path planning into two steps: first, using some common path planning method in a polygonal environment [6], then, considering UAV dynamic and kinematic constraints into the trajectory [14]. These methods assume decoupling, which affects the trajectory, as stated by all authors.

However, most of the effort focused on UAV trajectory planning is related to obstacle avoidance with kinodynamic constraints, without taking into account visibility analysis as part of the nature of the trajectory in urban environments.

The visibility problem has been extensively studied over the last twenty years, due to the importance of visibility in GIS and Geomatics, computer graphics and computer vision, and robotics. Accurate visibility computation in 3D environments is a very complicated task demanding a high computational effort, which could hardly have been done in a very short time using traditional well-known visibility methods [15]. The exact visibility methods are highly complex, and cannot be used for fast applications due to their long computation time. Previous research in visibility computation has been devoted to open environments using DEM models, representing raster data in 2.5D (Polyhedral model), and do not address, or suggest solutions for, dense built-up areas. Most of these works have focused on approximate visibility computation, enabling fast results using interpolations of visibility values between points, calculating point visibility with the Line of Sight (LOS) method [16]. Other fast algorithms are based on the conservative Potentially Visible Set (PVS) [17]. These methods are not always completely accurate, as they may render hidden objects' parts as visible due to various simplifications and heuristics.

A vast number of algorithms have been suggested for speeding up the process and reducing computation time. Franklin [18] evaluated and approximated visibility for each cell in a DEM model based on greedy algorithms. Wang et al. [19] introduced a Grid-based DEM method using viewshed horizon, saving computation time based on relations between surfaces and the LOS method. Later on, an extended method for viewshed computation was presented, using reference planes rather than sightlines [20].

One of the most efficient methods for DEM visibility computation is based on shadow-casting routine. The routine cast shadowed volumes in the DEM, like a light bubble [21]. Extensive research treated Digital Terrain Models (DTM) in open terrains, mainly Triangulated Irregular Network (TIN) and Regular Square Grid (RSG) structures. Visibility analysis in terrain was classified into point, line and region visibility, and several algorithms were introduced, based on horizon computation describing visibility boundary [22].

Only a few works have treated visibility analysis in urban environments. A mathematical model of an urban scene, calculating probabilistic visibility for a given object from a specific viewcell in the scene, has been presented by [23]. This is a very interesting concept, which extends the traditional deterministic visibility concept. Nevertheless, the buildings are modeled as cylinders, and the main challenges of spatial analysis and building model were not tackled. Other methods were developed, subject to computer graphics and vision fields, dealing with exact visibility in 3D scenes, without considering environmental constraints. Plantinga and Dyer [15] used the aspect graph – a graph with all the different views of an object. Due to their computational complexity, all of these works are not applicable to a large scene with near real-time demands, such as UAV trajectory planning.

III. VISIBILITY BOUNDARIES ANALYSIS

A. Problem Statement

We consider visibility problem in a 3D urban environment, consisting of static constant objects and dynamic objects.

Given:

- Static objects:
3D buildings modeled as 3D cubic parameterization

$$\sum_{i=1}^{N_{of_build}} C_i(x, y, z = \frac{h_{max}}{h_{min}})$$
- Dynamic objects:
Moving cars modeled as 3D cubic parameterization,

$$C_{car}(x, y, z)$$
- Pedestrian modeled as cylinder parameterization,

$$C_{peds}(x, y, z)$$
- Trees modeled with two cylinder parameterization,

$$C_{tree}(x, y, z)$$
- Wind profile $v_w(z)$.
- Viewpoint $V(x_0, y_0, z_0)$, in 3D coordinates.

Computes:

Set of all visible points from $V(x_0, y_0, z_0)$,

$$\sum_{i=1}^N [C_{building_i}, C_{car_i}, C_{tree_i}, C_{peds_i}]$$

We extend our previous work [2], developed for a fast and efficient visibility analysis for buildings in urban environments, and consider also a basic structure of cylinders, which allows us to model pedestrians and trees. Based on our probabilistic visibility computation of dynamic objects, we test the effect of these by using data gathered from web-oriented GIS sources to update our estimation and prediction on these entities.

B. Dynamic Objects – Modeling and Probabilistic Visibility

Dynamic objects such as moving cars and pedestrians, directly affect visibility in urban environments. Due to modeling limitations, these entities are usually neglected in spatial analysis aspects. We focus on three major dynamic objects in an urban case: moving cars and pedestrians. Each object is modeled with 3D boxes or 3D cylinders, which allow us to extend the use of our previous visibility analysis in urban environments presented for static objects [2].

1) Moving Car

3D Modeling: As we mentioned earlier, web-cameras in urban environments can record the moving cars at any

specific time. Image sources such as web cameras, like other similar sensors sources, demand an additional stage of Automatic Target Detection (ATD) algorithms to extract these objects from the image [31]. In this research we do not focus on ATD, which must be implemented when shifting from the research described in the paper toward an applicable system.

The common car structure can be easily modeled by two 3D boxes, as can be seen in Fig. 1(b), which is similar to the original car structure presented in Fig. 1(a).

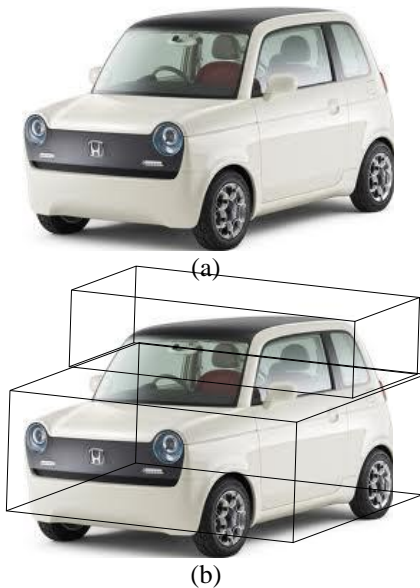


Fig. 1. Car Modeling Using 3D Boxes: (a) the Original Car, (b) the Modeled Car

We define the Car Boundary Points (CBP) as the set of visible surfaces' boundary points of 3D boxes modeling the car presented in Fig. 1(b). Each box is modeled as 3D cubic $C_{car}(x, y, z)$ as presented extensively in [2] for a building model case:

$$C_{car}(x, y, z) = \begin{pmatrix} x = t \\ y = \begin{pmatrix} x^n - 1 \\ 1 - x^n \end{pmatrix} \\ z = c \end{pmatrix}$$

$$\begin{aligned} -1 &\leq t \leq 1 \\ n &= 350 \\ c &= c + 1 \end{aligned}$$

Car Boundary Points (CBP) - we define CBP of the object i as a set of boundary points $j = 1..N_{CBP_bound}$ of the

visible surfaces of the car object, from viewpoint $V(x_0, y_0, z_0)$, where the maximum surface's number is six and each surface defined by four points, $N_{CBP_bound} \leq 24$.

In Fig. 2, the car is modeled by using two 3D boxes. Visible surfaces colored in red, CBP marked with yellow points.

$$CBP_{i=1..N_{CBP_bound}}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ \dots \\ x_{N_{CBP_bound}}, y_{N_{CBP_bound}}, z_{N_{CBP_bound}} \end{bmatrix}$$

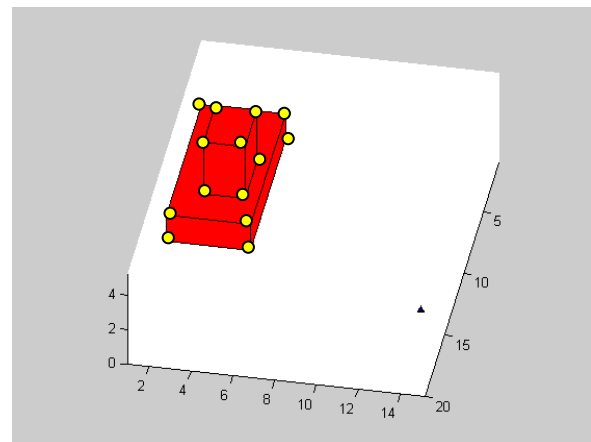


Fig. 2. Modeling Car Using 3D Boxes (CBP Marked with Yellow Points)

Probabilistic Visibility Analysis

Visibility has been treated as Boolean values. Due to incomplete information and the uncertainties of predicting the car's location at future times, visibility becomes much more complicated.

As it is well known from basic kinematics, CBP can be estimated in future time $t + \Delta t$ as:

$$CBP_i(t + \Delta t) = CBP_i(t) + V(t)\Delta t + \frac{A(t)\Delta t^2}{2}$$

Where $V(t)$ is the car velocity vector $V(t) = (v_x v_y)^T$, and the acceleration vector $A(t) = (a_x a_y)^T$. Estimation of a car's location in the future based on a web camera is not a simple task. Driver behavior generates multi-decision modeling, such as car-following behavior, gap acceptance behavior, or lane-change cases including traffic flow, speed etc. [32].

Our probabilistic car model is based on microscopic simulation models that were properly calibrated and validated using VISSIM simulation. VISSIM is a time-based microscopic simulation tool that uses various driver behaviors and vehicle performances to accurately represent

an urban traffic model. The VISSIM simulation model has been validated when compared to the data from various real-world situations [33] and used for the test-bed network by [34][35], and on driver behavior research defining average speed and acceleration [32].

The average speed in urban environments is about 45 [km/hr], from a minimum of 40 [km/hr] up to a maximum of 50 [km/hr]. In the situation of a free driving case, which is the common mode in urban environments [36], the acceleration of family car can change between 1 to 3.5 [m/sec²], and on average 2.5 [m/sec²], as seen in Fig. 3.

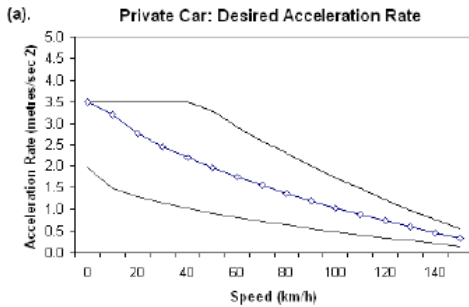


Fig. 3. Average Acceleration Rate of a Family Car in an Urban Environment [32]

As can be seen from several validations of car and driver estimation, velocity and acceleration are distributed as normal ones, and lead to normal location distribution:

$$V(t) \sim N(\mu = 45, \sigma^2 = 10)$$

$$A(t) \sim N(\mu = 2.5, \sigma^2 = 1)$$

$$CBP(t + \Delta t) \sim \sum N$$

In time step t , where the car's location is taken from a web-camera, visibility analysis from $CBP(t)$ is an exact one, based on our previous visibility analysis [2], as seen in Fig. 2. Visibility analysis becomes probabilistic for future time $t + \Delta t$, applying the same visibility analysis for $CBP(t + \Delta t)$ presented in Fig. 4.

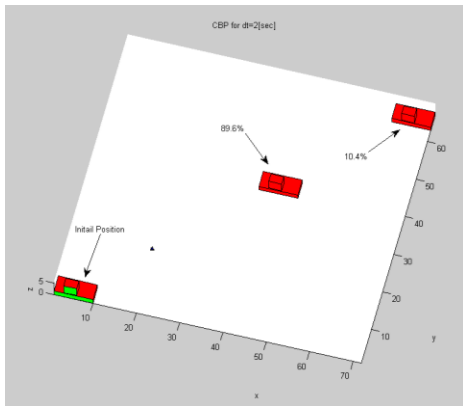


Fig. 4. Probabilistic Visibility Analysis for CBP

In Fig. 4, the car's location from a web-camera appears in the bottom left side. For $\Delta t = 2[\text{sec}]$, the car's location is marked by two 3D boxes, where CBP for each of them is the boundary of visible surfaces marked in red. The probability that the visible surfaces, which are bounded by CBP, will be visible in future time is based on the last update taken from the web application (depicted with arrows in Fig. 4, computed by using two different random normal PDF values for V and A based on eq. (4).

2) Pedestrians

3D Modeling: Pedestrian modeling can be done in high resolution, but due to ATD algorithms capabilities, pedestrians are usually bounded by a 3D cylinder and not as an exact detailed model [31]. For this reason, we model pedestrians as 3D cylinders, which is somewhat conservative but still applicable.

Pedestrian can be easily modeled by 3D cylinder, as seen in Fig. 5 (marked in red), which is similar to the output from ATD methods tested on a web-camera output recognizing walkers in urban environments.

We extend our previous visibility analysis concept [2] and include new objects modeled as cylinders as continuous curves parameterization $C_{Peds}(x, y, z)$.

Cylinder parameterization can be described as:

$$C_{Peds}(x, y, z) = \begin{pmatrix} r \sin(\theta) \\ r \cos(\theta) \\ c \end{pmatrix}$$

$$0 \leq \theta \leq 2\pi$$

$$c = c + 1$$

$$0 \leq c \leq h_{peds_max}$$

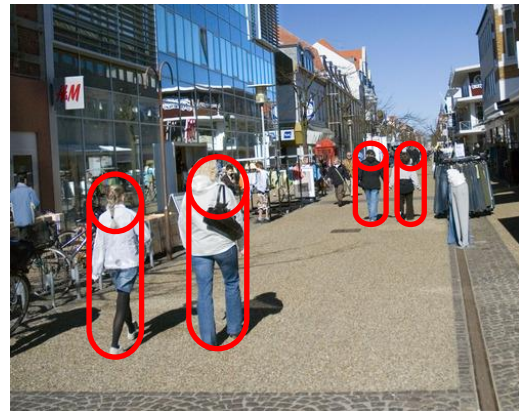


Fig. 5. Modeling Pedestrians in Urban Scene Using Cylinders (Colored in Red)

We define the visibility problem in a 3D environment for more complex objects as:

$$C'(x, y)_{z_{const}} \times (C(x, y)_{z_{const}} - V(x_0, y_0, z_0)) = 0$$

where 3D model parameterization is $C(x, y)_{z=const}$, and the viewpoint is given as $V(x_0, y_0, z_0)$. Extending the 3D cubic parameterization, we also consider the cylinder case. Integrating eq. (5) to (6) yields:

$$\begin{pmatrix} r \cos \theta \\ -r \sin \theta \\ 0 \end{pmatrix} \times \begin{pmatrix} r \sin \theta - V_x \\ r \cos \theta - V_y \\ c - V_z \end{pmatrix} = 0$$

$$\theta = \arctan \left(-\frac{-r - \frac{(-vy r + \sqrt{vx^4 - vx^2 r^2 + vy^2 vx^2})}{vx^2 + vy^2}}{vx} \right)$$

$$-\frac{-vy r + \sqrt{vx^4 - vx^2 r^2 + vy^2 vx^2}}{vx^2 + vy^2}$$

As can be noted, these equations are not related to Z axis, and the visibility boundary points are the same for each x-y cylinder profile.

The visibility statement leads to complex equation, which does not appear to be a simple computational task. This equation can be efficiently solved by finding where the equation changes its sign and crosses zero value; we used analytic solution to speed up computation time and to avoid numeric approximations. We generate two values of θ generating two silhouette points in a very short time computation. Based on an analytic solution to the cylinder case, a fast and exact analytic solution can be found for the visibility problem from a viewpoint.

We define the solution presented in eq. (8) as x-y-z coordinates values for the cylinder case as **Pedestrian Boundary Points** (PBP). PBP are the set of visible silhouette points for a 3D cylinder modeling the pedestrian, as presented in Fig. 6:

$$PBP_{i=1 \dots N_{PBP_bound}=2}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_{N_{PBP_bound}}, y_{N_{PBP_bound}}, z_{N_{PBP_bound}} \end{bmatrix}$$

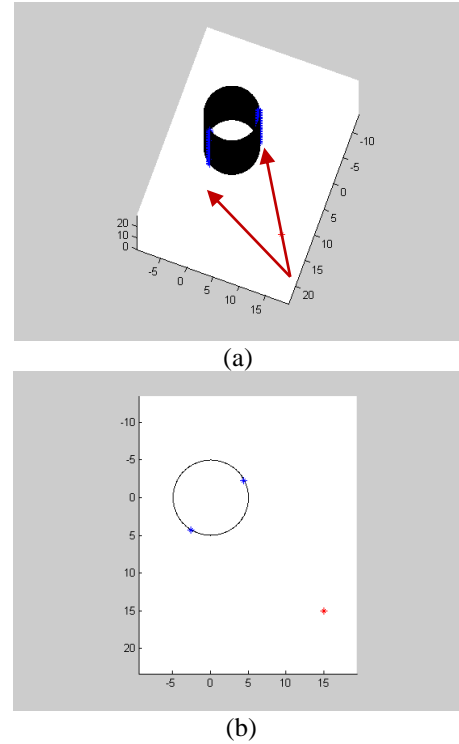


Fig. 6. PBP for a Cylinder using Analytic Solution marked as blue points, Viewpoint Marked in Red: (a) 3D View (Visible Boundaries Marked with Red Arrows); (b) Topside View

C. Visibility Analysis Considering Sensor's Stochastic Character

In this section, we extend our visibility model by exploring and including sensors' sensing capabilities and physical constraints. Our visibility analysis is based on the fact that sensors are located at specific visibility points. Sensors are commonly treated as deterministic detectors, where a target can only be detected or undetected. These simplistic sensing models are based on the disc model [37][38].

We study sensors' visibility-based placement effected by taking into account the stochastic character of target detection. We present a single sensor model, including noisy measurement, and define the necessary condition for visibility analysis with false alarm and detection probabilities for each visibility point's candidate.

1) Single Visibility Sensing Model

Most of the physical signals are based on energy vs. distance from single source model. Different kind of sensors such as: radars, lasers, acoustics, etc., are based on this signal character. Like other signal models presented in the literature [39][40][41] we use signal decay model as follows:

$$L(d) = \begin{cases} \frac{L_0}{(\frac{d}{d_0})^k}, & \text{if } d > d_0 \\ L_0, & \text{if } d \leq d_0 \end{cases}$$

where L_0 is the original energy emitted by the target, k is the decaying factor (typical values from 2 to 5), and d_0 is a constant determined by the size of the target and the sensor.

We model the sensor's noise N_i located at visibility point V_i , using zero-mean normal distribution, $N_i \sim N(0, \sigma^2)$. Sensor signal energy including noise effect, S_i , can be formulated as:

$$S_i = L(d_i) + N_i^2$$

In practice, S_i parameters are set by empiric datasets.

2) Necessary Condition for Visibility

Nowadays, detection systems use more and more data fusion methods [42][43]. In order to use multi sensors benefits, fusion and local decision-making using several sensors' data is a very common capability. As with other distributed data fusion methods, we assume that each sensor sends the energy measurement to a Local Decision Making Module (LDMM). Similar to other well known fusion methods [41], the LDMM integrates and compares the average sensors' measurements n against detection threshold τ .

Detection probability, denoted by P_D , is the probability that a target is correctly detected. Supposing that n sensors take part in the data fusion applied in the LDMM, detection probability is given by:

$$P_D = P\left(\frac{1}{n} \sum_{i=1}^n (L(d_i) + N_i^2) > \tau\right)$$

$$P_D = 1 - P\left(\sum_{i=1}^n \left(\frac{N_i}{\sigma}\right)^2 \leq \frac{n\tau - \sum_{i=1}^n L(d_i)}{\sigma^2}\right)$$

$$P_D = 1 - X_n\left(\frac{n\tau - \sum_{i=1}^n L(d_i)}{\sigma^2}\right)$$

Where $N_i/\sigma \sim N(0,1)$ and X_n denote the distribution function. In the same way, false alarm rate probability is the probability of making a positive detection decision when no target is present. False alarm rate probability, denoted by P_F , is given by:

$$P_F = P\left(\frac{1}{n} \sum_{i=1}^n N_i^2 > \tau\right) = 1 - P\left(\sum_{i=1}^n \left(\frac{N_i}{\sigma}\right)^2 \leq \frac{n\tau}{\sigma^2}\right)$$

$$P_F = 1 - X_n\left(\frac{n\tau}{\sigma^2}\right)$$

Conditions Necessary for Visibility: Given two real numbers, $a \in (0,1)$ and $b \in (0,1)$. Visibility Point

$V_i(x, y, z)$ can be defined as visible point **if and only if** $P_F(V_i) \leq a$ and $P_D(V_i) \geq b$.

We integrate our unique concept of probabilistic visibility into the velocity space. We transform the visibility's boundaries from location to velocity space.

IV. VISIBILITY VELOCITY OBSTACLES (VVO)

The visibility velocity obstacle represents the set of all velocities from a viewpoint, occluded with other objects in the environment. It essentially maps static and moving objects into the robot's velocity space considering visibility boundaries.

The VVO of an object with circular visibility boundary points such as the pedestrians case, PBP, that is moving at a constant velocity v_b , is a cone in the velocity space at point A. In Fig. 7, the position space and velocity space of A are overlaid to illustrate the relationship between the two spaces. The VVO is generated by first constructing the Relative Velocity Cone (RVC) from A to the boundaries of the object, i.e., PBP, then translating RVC by v_b .

Each point in VVO represents a velocity vector that originates at A. Any velocity of A that penetrates VVO is a occluded velocity that based on the current situation, would result in a occlusion between A and the pedestrian at some future time. Fig. 7 shows two velocities of A: one that penetrates VVO, hence a occluded velocity, and one that does not. All velocities of A that are outside of VVO are visible from the current robot's position as the obstacle denotes as B, stays on its current course. The visibility velocity obstacle thus allows determining if a given velocity is occluded, and suggesting possible changes to this velocity for better visibility. If PBP is known to move along a curved trajectory or at varying speeds, it would be best represented by the nonlinear visibility velocity obstacle case discussed next.

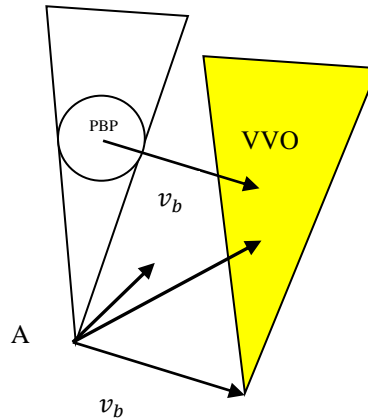


Fig. 7. Visibility Velocity Obstacles

The VVO consists of all velocities of A at t_0 predicting visibility's boundaries related to obstacles at the environment at any time $t > t_0$. Selecting a single velocity, v_a , at time $t = t_0$ outside the VVO, guarantees visibility to this specific obstacle at time t . It is constructed as a union of its temporal elements, $VVO(t)$, which is the set of all absolute velocities of A, v_a , that would allow visibility at a specific time t . Referring to Fig. 8, v_a that would result in occlusion with point p in B at time $t > t_0$, expressed in a frame centered at $A(t_0)$, is simply:

$$v_a = \frac{VBP_i}{t - t_0}$$

where r is the vector to point p in the blocker's fixed frame, and visibility boundaries denoted as Visibility Boundary Points (VBP). The set, $VVO(t)$ of all absolute velocities of A that would result in occlusion with any point in B at time $t > t_0$ is thus:

$$VVO(t) = \frac{VBP_i(t)}{t - t_0}$$

Clearly, $VVO(t)$ is a scaled B for two dimensional case with circular object, located at a distance from A that is inversely proportional to time t . The entire VVO is the union of its temporal subsets from t_0 , the current time, to some set future time horizon t_h :

$$VVO(t) = \bigcup_{t=t_0}^{t_h} \frac{VBP_i(t)}{t - t_0}$$

The presented VVO generate a warped cone in a case of 2D circular object. If $VBP(t)$ is bounded over $t = (t_0, \infty)$, then the apex of this cone is at $A(t_0)$. We extend our analysis to 3D general case, where the objects can be cubes, cylinders and circles. The mathematical analysis with visibility boundaries is based on VBP presented in the previous part for different kind of objects such as buildings, cars and pedestrians.

We transform the visibility's boundaries into the velocity space, by moving the VBP to the velocity space, in the same analysis presented for 2D circle boundary's. Following that, we present 3D extension for VBP case, transformed to the velocity space.

Given two objects, VBP_1 , VBP_2 will create a VVO representing VBP_2 (and vice-versa) such that VBP_1 wishes to choose a guaranteed collision-free velocity for the time interval τ , and visibility boundary in velocity space.

The Nonlinear Visibility Velocity Obstacle (NVVO) accounts for a general trajectory of the object, while assuming a constant velocity of the robot. It applies to the scenario shown in Fig. 8, where, at time t_0 , a point A attempts to plan visible trajectory related an object, PBP, that is following a general known trajectory, $c(t)$, and at time t_0 is

located at $c(t_0)$. PBP represents the set of points that define the geometry of the visibility boundaries of the object, grown by the radius of the robot. In case of pedestrians where PBP is a circle, then $c(t)$ represents the trajectory followed by its center.

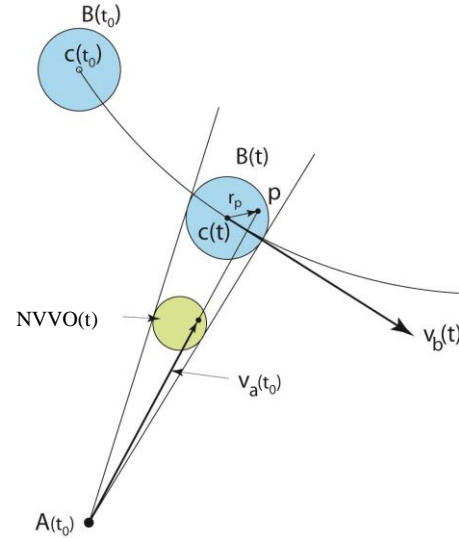


Fig. 8. Nonlinear Visibility Velocity Obstacles (NVVO), based on Nonlinear Velocity Obstacles (NLVO) (source [44])

Our method, based on visibility boundaries transformation from position to velocity space, can be formulated as homothetic transformation [44] that is centered at $A(t_0)$ and having the ratio $k = 1/(t - t_0)$:

$$v_a = H_{A(t_0),k}(c(t) + r), k = \frac{1}{t - t_0}$$

The set, $NVVO(t)$ of all absolute velocities of A that would result in occlusion with objects B at time $t > t_0$ is thus:

$$NVVO(t) = H_{A(t_0),k}(c(t) \oplus B), k = \frac{1}{t - t_0},$$

where \oplus represents the Minkowski sum. Clearly, $NVVO(t)$ is a scaled B, located at a distance from A that is inversely proportional to time t . To emphasize the geometric shape of the $NVVO(t)$, we rewrite it as:

$$NVVO(t) = \frac{c(t)}{t - t_0} \oplus \frac{B}{t - t_0}$$

The entire NVVO is the union of its temporal subsets from t_0 , the current time, to some set time horizon t_h :

$$NVVO = \bigcup_{t_0 < t < t_h} \frac{c(t)}{t - t_0} \oplus \frac{B}{t - t_0}$$

Truncating the NVVO at t_h allows focusing the analysis till limited future time, time horizon. In case of cars, buildings and pedestrians where visibility boundaries can be expressed by geometric operations of 3D boxes, where VVO for the linear and NVVO for the non linear case analyzed in the same concept and formulation presented so far, as can be seen in Fig. 9.

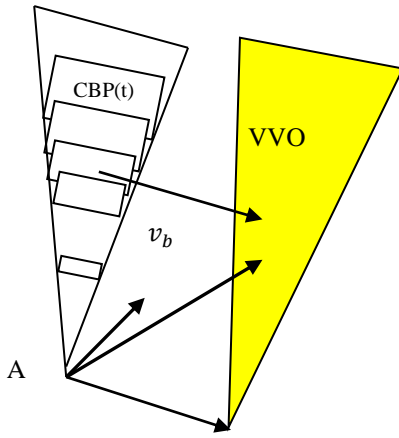


Fig. 9. Visibility Velocity Obstacle for visibility boundaries consist of 3D boxes

V. PURSUER PLANNER USING VVO

Our planner, similar to previous work [45] is a local one, generating one step ahead every time step reaching toward the goal, which is a depth first A* search over a tree. We extend previous planners, which take into account kinematic and dynamic constraints [9][14] and present a local planner for UAV as case study with these constraints, which for the first time generates fast and exact visible trajectories based on VVO, tracking after a target by choosing the optimal next action based on velocity estimation.

The fast and efficient visibility analysis of our method, allows us to generate the most visible trajectory from a start state q_{start} to the goal state q_{goal} in 3D urban environments, which can be extended to real performances in the future. We assume knowledge of the 3D urban environment model, and by using Visibility Velocity Obstacles (VVO) method

to avoid occlusion, exploring maximum visible node in the next time step and track a specific target.

At each time step, the planner computes the next eighth Attainable Velocities (AV), as detailed in the next sub-section. The nodes, which are not occluded, i.e., nodes outside Visibility Velocity Obstacles, are explored. The planner computes the cost for these visible nodes and chooses the node with the optimal cost according to mission type. In our case, the optimal cost related to the node with minimum velocities difference between the robot and the tracked target.

1) Attainable Velocities

Based on the dynamic and kinematic constraints, UAVs velocities at the next time step are limited. At each time step during the trajectory planning, we map the AV, the velocities set at the next time step $t + \tau$, which generate the optimal trajectory, as is well-known from Dubins theory [28].

We denote the allowable controls as $u = (u_s, u_z, u_\phi)$ as U , where $V \in U$.

We denote the set of dynamic constraints bounding control's rate of change as $\dot{u} = (\dot{u}_s, \dot{u}_z, \dot{u}_\phi) \in U'$.

Considering the extremal controllers as part of the motion primitives of the trajectory cannot ensure time-optimal trajectory for Dubins airplane model [28], but is still a suitable heuristic based on time-optimal trajectories of Dubin - car and point mass models.

We calculate the next time step's feasible velocities $\tilde{U}(t + \tau)$, between $(t, t + \tau)$:

$$\tilde{U}(t + \tau) = U \cap \{u \mid u = u(t) \oplus \tau \cdot U'\}$$

Integrating $\tilde{U}(t + \tau)$ with UAV model yields the next eight possible nodes for the following combinations:

$$\tilde{U}(t + \tau) = \begin{pmatrix} \tilde{U}_s(t + \tau) \\ \tilde{U}_z(t + \tau) \\ \tilde{U}_\phi(t + \tau) \end{pmatrix} = \begin{pmatrix} u_s^{\min}(t) + a_s \tau \\ -u_s^{\max} \tan \phi^{\max}, u_s(t) \tan u_\phi(t) + u_s^{\max} \tan a_\phi \\ u_z^{\max}, u_z(t) - a_z \tau \end{pmatrix}$$

At each time step, we explore the next eight AV at the next time step as part of our tree search, as explained in the next sub-section.

2) Tree Search

Our planner uses a depth first A* search over a tree that expands over time to the goal. Each node (q, \dot{q}) , where

$q = (x, y, z, \theta)$, consist of the current UAVs position and velocity at the current time step. At each state, the planner computes the set of AV, $\tilde{U}(t + \tau)$, from the current UAV velocity, $U(t)$, as shown in Fig. 10. We ensure the visibility of nodes by computing a set of Visibility Velocity Obstacles (VVO).

In Fig. 10, nodes inside VVO, marked in red, are occluded. Nodes out of VVO are further evaluated; visible nodes are colored in blue. The safe node with the lowest cost, which is the next most visible node, is explored in the next time step. This is repeated while generating the most visible trajectory, as discussed in the next sub-section.

Attainable velocities profile is similar to a trunked cake slice, as seen in Figure 10, due to the Dubins airplane model with one time step integration ahead. Simple models attainable velocities, such as point mass, create rectangular profile [4].

3) Cost Function

Our search is guided by minimum invisible parts from viewpoint V to the 3D urban environment model, with minimal difference between robot's velocity v_a and tracked target v_{tck} .

The cost function is computed for each visible node $(q, \dot{q}) \in VVO$, i.e., node outside VVO, considering UAV velocities at the next time step:

$$w(q(t + \tau)) = \text{abs}(v_a(q(t + \tau)) - v_{tck}(q(t + \tau)))$$

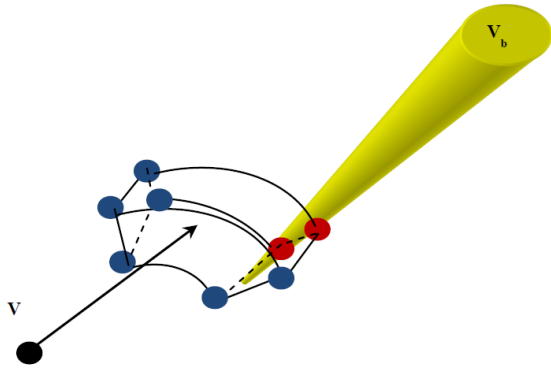


Fig. 10. Tree Search Method

4) Planner Pseudo-Code

The Pseudo-Code of the UAV Planner is as follows in Fig. 11:

```

 $t = t_0$ .  $q_{best} = q_{start}$ 
1. While ( $q_{best} \neq q_{goal}$ ) do:
    1.1. Calculate AV nodes from  $q_{best}$ ,
     $AV_{i=1}^8 = U_{i=1}^8(t + \tau)$ .
    1.2. For each node  $q_i \in AV$  check:
        if  $\dot{q}_i \in \bigcup_{j=1}^{n=N_{onj}} VVO_j$ 
             $q_i$  is illegal.
        Else
            Calculate node cost  $w(q_i)$ 
    1.3. If all nodes are illegal
        STOP! No trajectory to the goal
    Else
        1.3.1. Find node with minimal cost
         $q_{min} = \{q_i \mid \min w(q_i)\}$ .
        1.3.2. Update  $q_{best} = q_{min}$ 
        1.3.3.  $t = t + dt$ 
    End

```

Fig. 11. UAV Planner Pseudo-Code

VI. CONCLUSIONS

This paper proposes an online motion planning algorithm in 3D environments for tracking target, taking into account visibility analysis. The planner is based on local search and includes dynamic and kinematic constraints as complete part of the planner. Visibility boundaries, which are based on analytic solution for several kind of objects in 3D urban environments, also include uncertainty and probabilistic factors. Each VVO represents velocity's set of possible future collision and visibility boundaries. Based on our analysis in velocity space, we plan our trajectory by selecting future robot's velocity at each time step, tracking after specific target considering visibility constraints as integral part of the velocities space. We formulate the tracked target in the environment and include visibility analysis for the next time step as part of our planning in the same search space.

REFERENCES

- [1] O. Gal and Y. Doytsher, "Motion Planning in 3D Environments Using Visibility Velocity Obstacles," in Proc. of the Tenth International Conference on Advanced Geographic Information Systems, Applications, and Services, Athens, Greece, pp: 60-65, 2018
- [2] O. Gal and Y. Doytsher, "Fast and Accurate Visibility Computation in a 3D Urban Environment," in Proc. of the

- Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain, pp: 105-110, 2012
- [3] O. Gal and Y. Doytsher, "Fast Visibility Analysis in 3D Procedural Modeling Environments," in Proc. of the, 3rd International Conference on Computing for Geospatial Research and Applications, Washington DC, USA, 2012
 - [4] P. Fiorini and Z. Shiller, "Motion Planning in Dynamic Environments Using Velocity Obstacles," *Int. J. Robot. Res.* 17, pp. 760-772, 1998
 - [5] Office of the Secretary of Defense, Unmanned Aerial Vehicles Roadmap, Tech. rep., December 2002
 - [6] J.C. Latombe, "Robot Motion Planning," Kluwer Academic Press, 1990
 - [7] M. Erdmann and T. Lozano-Perez, "On Multiple Moving Objects," *Algorithmica*, 2, 477-521, 1987
 - [8] T. Fraichard, "Trajectory Planning in a Dynamic Workspace: A 'State-Time Space' Approach," *Advanced Robotics*, 13:75-94, 1999
 - [9] S.M. LaValle and J. Kuffner, "Randomized Kinodynamic Planning," In Proc. IEEE Int. Conf. on Robotics and Automation, Detroit, MI, USA, pp: 473-479, 1999
 - [10] Z.H. Mao, E. Feron, and K. Bilimoria, "Stability and Performance of Intersecting Aircraft Flows Under Decentralized Conflict Avoidance Rules," *IEEE Transactions on Intelligent Transportation Systems*, 2: 101-109, 2001
 - [11] J. Bellingham, A. Richards, and J. How, "Receding Horizon Control of Autonomous Aerial Vehicles," in Proceedings of the IEEE American Control Conference, Anchorage, AK, USA, pp. 3741-3746, 2002
 - [12] B. Sinopoli, M. Micheli, G. Donata, and T. Koo, "Vision Based Navigation for an Unmanned Aerial Vehicle," in Proc. IEEE Int'l Conf. on Robotics and Automation, 2001
 - [13] J. Sasiadek and I. Duleba, "3D Local Trajectory Planner for UAV," *Journal of Intelligent and Robotic Systems*, 29: 191-210, 2000
 - [14] S.A. Bortoff, "Path Planning for UAVs," In Proc. of the American Control Conference, Chicago, IL, USA, pp: 364-368, 2000
 - [15] H. Plantinga and R. Dyer, "Visibility, Occlusion, and Aspect Graph," *The International Journal of Computer Vision*, 5, 137-160, 1990
 - [16] Y. Doytsher and B. Shmutter, "Digital Elevation Model of Dead Ground," Symposium on Mapping and Geographic Information Systems (Commission IV of the International Society for Photogrammetry and Remote Sensing), Athens, Georgia, USA, 1994
 - [17] F. Durand, "3D Visibility: Analytical Study and Applications," PhD thesis, Universite Joseph Fourier, Grenoble, France, 1999
 - [18] W.R. Franklin, "Siting Observers on Terrain," in Proc. of 10th International Symposium on Spatial Data Handling. Springer-Verlag, pp. 109-120, 2002
 - [19] J. Wang, G.J. Robinson, and K. White, "A Fast Solution to Local Viewshed Computation Using Grid-based Digital Elevation Models," *Photogrammetric Engineering & Remote Sensing*, 62, 1157-1164, 1996
 - [20] J. Wang, G.J. Robinson, and K. White, "Generating Viewsheds without Using Sightlines," *Photogrammetric Engineering & Remote Sensing*, 66, 87-90, 2000
 - [21] C. Ratti, "The Lineage of Line: Space Syntax Parameters from the Analysis of Urban DEMs," *Environment and Planning B: Planning and Design*, 32, 547-566, 2005
 - [22] L. De Floriani and P. Magillo, "Visibility Algorithms on Triangulated Terrain Models," *International Journal of Geographic Information Systems*, 8, 13-41, 1994
 - [23] B. Nadler, G. Fibich, S. Lev-Yehudi, and D. Cohen-Or, "A Qualitative and Quantitative Visibility Analysis in Urban Scenes," *Computers & Graphics*, 5, 655-666, 1999
 - [24] S.M. LaValle, "Planning Algorithms," Cambridge, U.K.: Cambridge Univ. Pr., 2006
 - [25] M. Hwangbo, J. Kuffner, T. Kanade, "Efficient Two-phase 3D Motion Planning for Small Fixed-wing UAVs," In proceeding of: 2007 IEEE International Conference on Robotics and Automation, ICRA 2007, 10-14 April 2007, Roma, Italy
 - [26] <http://www.asctec.de/uav-applications/research/products/asctec-hummingbird/>
 - [27] A. Bhatia, M. Graziano, S. Karaman, R. Naldi, E. Frazzoli, "Dubins Trajectory Tracking using Commercial Off-The-Shelf Autopilots," AIAA Guidance, Navigation and Control Conference and Exhibit 18 - 21 August 2008, Honolulu, Hawaii.
 - [28] H. Chitsaz and S.M. LaValle, "Time-Optimal Paths for a Dubins Airplane," in Proc. IEEE Conf. Decision and Control, USA, pp. 2379-2384, 2007
 - [29] S. Zlatanov, A. Rahman, and S. Wenzhong, "Topology for 3D Spatial Objects," *International Symposium and Exhibition on Geoinformation*, pp. 22-24, 2002
 - [30] W.R. Franklin and C. Ray, "Higher isn't Necessarily Better: Visibility Algorithms and Experiments," In T. C. Waugh & R. G. Healey (Eds.), *Advances in GIS Research: Sixth International Symposium on Spatial Data Handling*, pp. 751-770. Taylor & Francis, Edinburgh, 1994
 - [31] Y. Song, "The research of a new Auto Target Recognition directed Image compression," in 3th Int. Congress on Image and Signal Processing (CISP), 16-18 Oct, China, 2010
 - [32] J. Archer, "Methods for the Assessment and Prediction of Traffic Safety at Urban Intersections and their Application in Micro-simulation Modeling," Centre for Traffic Simulation Research, CTR, Sweden. Technical Report, 2010
 - [33] M. Fellendorf and P. Vortisch, "Validation of the Microscopic Traffic Flow Model VISSIM in Different Real world Situations," 79th Annual meeting of Transportation Research Board, UK, 2001
 - [34] B. Park and J. D. Schneeberger, "Microscopic Simulation Model Calibration and Validation: Case Study of VISSIM Simulation Model for a Coordinated Actuated Signal System," *Transportation Research Record* 1856, Paper No. 03-2531
 - [35] D. Parker and T. Lajunen, "Are Aggressive People Aggressive Drivers? A Study of the Relationship between Self-Reported General Aggressiveness Driver Anger and Aggressive Driving," *Accident Analysis and Prevention*, 33(2), 243-255, 2001
 - [36] R. Wiedemann and U. Reiter, "Microscopic Traffic Simulation: The Simulation System MISSION," Background and Actual State. Project ICARUS (V1052), Final Report, Brussels CEC.2: Appendix A, 1992
 - [37] K. Chakraborty, S. Iyengar, H. Qi and E. Cho, "Grid Coverage for Surveillance and Target Location in Distributed Sensor Networks," *IEEE Trans. Comput.*, vol. 51, no. 12, 2002
 - [38] D. Tian and N.D. Georganas, "A coverage-preserved node scheduling scheme for large wireless sensor networks," In *WSNA*, 2002
 - [39] M.F. Duarte and Y.H. Hu, "Vehicle classification in distributed sensor networks," *Journal of Parallel and Distributed Computing*, vol. 64, no. 7, 2004
 - [40] D. Li and Y.H. Hu, "Energy based collaborative source localization using acoustic micro-sensor array," *EUROSIP J. Applied Signal Processing*, vol. 4, 2003
 - [41] P. Varshney, "Distributed Detection and Data Fusion," Springer-Verlag, 1996

- [42] T. Clouqueur, V. Phipatanasuphorn, P. Ramanathan and K.K. Saluja, "Sensor deployment strategy for target detection," In WSNA, 2002
- [43] T. Clouqueur, K.K. Saluja and P. Ramanathan, "Fault tolerance in collaborative sensor networks for target detection," IEEE Trans. Comput, vol. 53, no. 3, 2004
- [44] Z. Shiller, R. Prasanna, J. Salinger, "A Unified Approach to Forward and Lane-Change Collision Warning for Driver Assistance and Situational Awareness," SAE Technical Paper 2008-01-0204, 2008, <https://doi.org/10.4271/2008-01-0204>
- [45] O. Gal and Y. Doytshe. "Patrolling Strategy Using Heterogeneous Multi Agents in Urban Environments Using Visibility Clustering", Journal of Unmanned System Technology, ISSN 2287-7320, 2016

A Comprehensive Workplace Environment based on a Deep Learning Architecture for Cognitive Systems

Using a multi-tier layout comprising various cognitive channels, reflexes and reasoning

Thorsten Gressling

ARS Computer und Consulting GmbH
Munich, Germany
e-mail: thorsten.gressling@ars.de

Veronika Thurner

Munich University of Applied Sciences
Department of Computer Science and Mathematics
Munich, Germany
e-mail: veronika.thurner@hm.edu

Abstract—Many technical work places, such as laboratories or test beds, are the setting for well-defined processes requiring both high precision and extensive documentation, to ensure accuracy and support accountability that often is required by law, science, or both. In this type of scenario, it is desirable to delegate certain routine tasks, such as documentation or preparatory next steps, to some sort of automated assistant, in order to increase precision and reduce the required amount of manual labor in one fell swoop. At the same time, this automated assistant should be able to interact adequately with the human worker, to ensure that the human worker receives exactly the kind of support that is required in a certain context. To achieve this, we introduce a multilayer architecture for cognitive systems that structures the system's computation and reasoning across well-defined levels of abstraction, from mass signal processing up to organization-wide, intention-driven reasoning. By partitioning the architecture into well-defined, distinct layers, we reduce complexity and thus facilitate both the implementation and the training of the cognitive system. Each layer comprises a building block that adheres to a specific structural pattern, consisting of storage, processing units and components that are used for training. We incorporate ensemble methods to allow for a modular expansion of a specific layer, thus making it possible to introduce pre-trained functional blocks into the system. In addition, we provide strategies for generating synthetical data that support the training of the neural network parts within the processing layers. On this basis, we outline the functional modules of a cognitive system supporting the execution of partially manual processes in technical work places. Finally, a prototypical implementation serves as a proof of concept for this multilayer architecture.

Keywords—Cognitive system; Multilayer architecture; Technical work place; Machine learning; Ensemble averaging; Synthesized training data; Context sensitive; Neural network.

I. MOTIVATION

Many technical work places, such as laboratories or test beds, are the setting for series of well-defined, repetitive actions requiring high precision in their execution, as well as extensive documentation of every process step. Both are necessary to ensure accuracy on the one hand, and on the other hand to support accountability that often is required by law, science, or both. Typical examples are laboratories for micro-biological analysis or chemical experiments, premises of optometrists or hearing aid acousticians, or test beds for the quality inspection of produced goods, such as measuring vehicle exhaust fumes or assessing nutritional values of food.

All these settings share a number of commonalities. For one thing, within each of these working settings a human being interacts extensively with technical devices, such as measuring instruments or sensors. For another thing, processing follows a well-defined routine, or even precisely specified interaction protocols. Finally, to ensure that results are reproducible, the different steps and achieved results usually have to be documented extensively and in a precise way.

Especially in scenarios that execute a well-defined series of actions, it is desirable to delegate certain routine tasks, such as documentation or preparatory next steps, to some sort of automated assistant, in order to increase precision and reduce the required amount of manual labor in one fell swoop. At the same time, this automated assistant should be able to interact adequately with the human worker, to ensure that the human worker receives exactly the kind of support that is required in a certain context. To achieve this, the assistant needs to be context aware, i.e., equipped with cognitive input channels. As well, the assistant is expected to learn new behavioral patterns from previous experience. Thus, training the assistant appropriately is highly relevant for ensuring that the assistant's contribution really is beneficial to the human worker that it supports.

Traditional software systems for process control or workflow management are well able to support a set of well-defined processes that has been explicitly specified in advance. However, in situations that were not foreseen initially, or that were not explicitly specified because they were deemed to be highly unlikely to happen, these systems quickly meet their limits, requiring human take-over and problem solving abilities in expert mode.

The increasing capabilities of cognitive systems imply the potential for a new generation of systems that offer context sensitive reasoning on a scale hitherto unknown in machines and software systems. We exploit these possibilities by devising a cognitive hardware-software-system for supporting the execution of hybrid (i.e., partly manual and partly automated) processes in technical environments, in a manner that combines the respective strengths of human-expert-like cognition and reasoning with machine-like processing power, to improve performance, efficiency, accuracy and security issues.

By *cognitive system*, we denote a system that is capable of sensory perception and of expressing itself via technical devices, analogously to corresponding abilities found in

biological organisms. Furthermore, it comprises an internal representation that is comparable to emotions. However, as system boundaries and internal states of an artificial intelligence differ greatly from those of humans beings and animals, its underlying system of values and beliefs in general differs from that of biological organisms.

After this initial motivation, we review related literature and briefly sketch the goals of our research in Section II. In Section III, we introduce the physical architecture of our cognitive system, followed by a logical architecture in Section IV. The core element of the logical architecture are building blocks, whose structure is presented in Section V. Section VI discusses approaches for effectively training the system. To illustrate the processing of our cognitive system and the interaction of the building blocks on the different layers, we present an example execution in Section VII. Section VIII sketches the prototypical implementation that we realized as a proof of concept. Finally, we critically discuss our work in Section IX, before summarizing it in Section X.

II. STATE OF THE ART AND GOALS

Research has already elicited several aspects that are relevant in this context. In [1], an initial version of a multilayer architecture for cognitive systems is introduced, which is enhanced here by insights into the training of the different layers that the architecture comprises. As well, the usage of ensembles is considered here, in order to improve training performance and prepare for neural reasoning.

An overview of existing approaches to computation and information architecture is provided by [2], distinguishing among others the different types of information that are processed, from subsymbolic computation focusing on data and signal processing to symbolic computation that processes data structures, thus reflecting different levels of abstraction. In [3], patterns for cognitive systems are investigated into, focusing on systems that process textual information, yet indicating that other kind of information, such as cognitively interpreted sensory data, will be addressed by cognitive systems in the near future, thus entering into new dimensions of machine cognition.

Approaches for systematic process support through Context Aware Assistive Systems (CAAS) are discussed in [4] [5] [6], in the context of manufacturing on the shop floor and human interaction with the production line. Identifying human actions observed via cameras and relating them to the manufacturing process are a crucial issue in these Context Aware Assistive Systems.

The research from [7] [8] [9] focusses on the usage of augmented reality in intelligent assistant systems, discussing among others digital projections into the current working situation, to guide the human workers through their share of the working process.

Anderson et al. [10] introduce the cognitive architecture ACT-R, which implements artificial intelligence in a symbolic way. In contrast to this, in our approach we combine symbolic and connectionistic aspects.

So far, existing supportive systems for processes that are partially executed manually in technical work places are realized mainly in a rule oriented way and implemented by algorithms. As a consequence, they cover only those situations,

states and actions that have been anticipated in advance. However, they only have limited ability to learn from experience.

Recently, research on context awareness significantly progressed towards identifying a specific situation from a predefined set of possibilities in a given context and well-defined surroundings, incorporating cognition and artificial reasoning on a single level of abstraction. This single level of abstraction is then realized as a monolithic block of neural networks. However, this monolithic block needs to deal with the entire complexity by itself, which would require an extreme amount of training that exceeds what can be handled even by modern hardware.

Therefore, as a next step, we introduce an architecture for cognitive systems that expands cognition and reasoning across several levels of abstraction, to support a wide range of assistant services ranging from small actions to strategic processes. By partitioning the systems's cognition and reasoning into separate levels of abstraction (rather than implementing them as a single monolithic block), we reduce both implementation complexity and training effort. Thus, it is possible to tackle even very complex problems, which would exceed the capacity of a monolithic approach. As well, each building block of the architecture comprises components that are designated for training purposes, i.e., for generating synthetical training data and using this data to calibrate the neural network parts of the processing component.

We discuss the applicability of our approach in the context of a cognitive system that supports hybrid processes involving manual tasks within technical surroundings.

III. PHYSICAL ARCHITECTURE

Physically, a technical work place comprises a variety of technical devices, as a relevant tool set to execute, or support the execution of, actions involved in the processes at the work place. Typical examples for, e.g., a chemical laboratory are electronic high precision scales, a centrifuge, a power supply or a fume hood. Some of these devices are connected to computational hardware (e.g., a remote server), either directly or via a data network. In contrast to this, other devices such as a traditional heater, operate in an isolated way, without any direct data exchange with the computational hardware. Furthermore, the work place comprises a variety of tools and devices for manual tasks, such as pipettes or glassware, as well as other materials, e.g., chemical or biological substances to be processed or analyzed.

In addition, to evolve from a technology interspersed work place towards an intelligent assistant, the work place must be equipped with devices that enable the system's cognition, as well as its interaction with the human user that executes the manual process steps. Traditionally, cognitively exploitable input devices are, for example, a microphone (with subsequent speech analysis) or a camera (with subsequent image processing). In addition to this traditional notion of audio-visual cognition, sensors and other technical measuring devices provide additional cognitive channels that supply the system with information on the current situation at the work place.

Communication from the system towards the human user is realized, e.g., via a monitor, a loudspeaker or a projector that focusses its beam of light on the tool to be used next, or that displays instructions on the process step that should

be executed next. Other, more sophisticated devices arise continuously, such as mixed reality smart glasses for displaying instructions directly into the field of vision, activity tracking bracelets that combine skin and body sensors with functionality for alerting its wearer, or even EEGs for integrating information on the human user's brain activity into the system's data pool.

Note that some of these technical devices may include their own data storage, as well as computational hardware, thus being able to directly aggregate and process the data they collected, before passing it on to more sophisticated computational hardware for integration with the data from other devices and subsequent further processing.

IV. LOGICAL ARCHITECTURE

The cognitive system that we devise to support process execution is embedded into this technical work place.

Logically, we design a multilayer architecture that structures the cognitive system into different levels of abstraction (see Figure 1), rising from concrete at the bottom towards more and more abstract as we move upwards on the processing level stack. Thus, each layer M_j encapsulates processing on a specific level of abstraction, and focuses on different tasks. By structuring the overall system into logical processing layers, it is possible to train each layer individually for its respective tasks. Furthermore, modularizing the overall system improves performance by reducing processing time, as the different layers can be run in parallel.

Processing involves the analysis of incoming data, which is synthesized and analyzed to identify the situation that the work place is in, corresponding to an overall system state in the context of the executed processes. From the identified situation, processing derives, which actions should be taken as next steps, and passes these on as instructions to other layers, systems, technical devices or – via output devices – to the human user. Thus, the cognitive system is able to effectively support the human user in a context sensitive way. Note that these suggested actions are determined by aggregated conclusions that the system draws from its analysis.

Layers are interconnected by communication channels. Note that the information on situations flows upwards in the processing layer stack via channels S (white block arrows in Figure 1), whereas instructions are passed down from layer to layer via channels I (black block arrows).

The processing layer stack is based on a layer of technical devices D_1, \dots, D_m as described above. These devices collect data on the work place and enter these into the cognitive system as situation information via channels $S_{1..i}$, with $1 \leq i \leq m$. Some of these devices (such as D_m in Figure 1) merely collect data, e.g., by simple measuring, and pass them straight on to the first processing layer M_1 . Other devices (such as D_1 and D_2 in Figure 1) comprise an independent processing component ($M_{0.1}$ or $M_{0.2}$, respectively), which preprocesses the data before entering it into the first processing layer.

Moving upwards on the processing layer stack, situation information is aggregated from separate small snippets of measured data into larger contexts, such as actions, sequences of actions or even entire processes. Analogously, abstract instructions that are passed from top to bottom are made more

and more specific from layer to layer, down to signals that operate a specific technical device in the bottom layer.

On each layer, processing takes into account the situation information that is entered into the layer from below, as well as the instruction information that is passed to the layer from above. Thus, situations are interpreted in the light of instructions that reflect the larger context of the overall system, as identified on the higher levels of abstraction.

As depicted for layer M_j in Figure 1, a processing layer can merge several process layer stacks, each representing a different work place. Thus, their information flows are integrated and consolidated, allowing for integrated information processing on a cross-organizational level of abstraction.

V. BUILDING BLOCKS

Each layer in Figure 1 is implemented by a building block that follows the architectural pattern depicted in Figure 2 for a building block i , with $0 \leq i \leq n$, in a cognitive system that comprises $n \geq 1$ processing layers stacked on top of one layer of technical devices.

Building block i is linked with the building blocks of its surrounding processing levels $i-1$ and $i+1$ via communication channels, depicted as block arrows in Figure 2. Thus, the situation information perceived by building block $i-1$ is passed on via channel S_i to building block i , which stores the information in its storage for situations. Analogously, instructions issued by building block $i+1$ are passed on via channel $I(i+1)$ to building block i , which stores the information in its storage for instructions.

The central part of each building block is its processing unit, which comprises both aspects of algorithmic logic and of artificial intelligence (implemented via one or more neural networks), in varying proportions (see Figure 3). On the lower levels of abstraction, the major part of processing is accomplished by algorithmic logic, whereas on the higher levels of abstraction, aspects of artificial intelligence dominate the processing.

The processing unit works on three different kinds of input:

- $E1$: Information on situations
- $E2$: Information on instructions
- $E3$: Relevant general factual knowledge, stored as rules in the knowledge database

Based on these inputs, the processing unit analyses the incoming information, interprets it and synthesizes it into its interpretation of the situation, thus lifting the previous information on the situation onto a higher level of abstraction. For example, on layer M_1 , short snippets of data that were measured by the technical devices (e.g., an electronic scale and a heater) in the underlying physical layer are gathered, consolidated and then passed on to layer M_2 . On layer M_2 , this consolidated data is then merged and interpreted to build a larger semantic context, e.g., a certain step in a chemical experiment where a certain amount of substance must be added to an existing mixture, and then heated to a specific temperature. In order to properly identify the semantic context correctly, the processing unit in layer M_2 incorporates known "recipes" of chemical experiments that are stored within the knowledge database of layer M_2 .

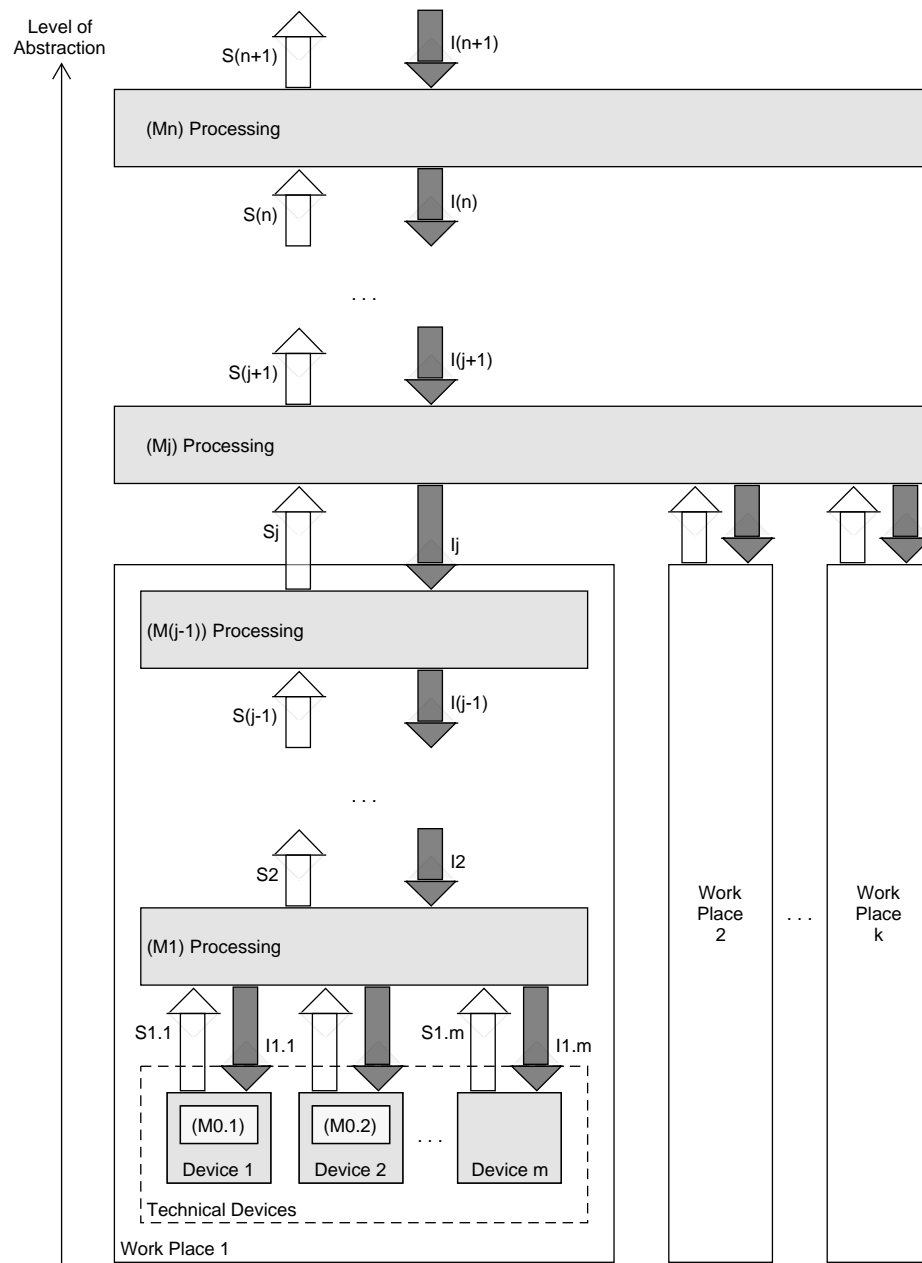


Figure 1. Multilayer architecture of cognitive systems for supporting hybrid processes in technical work places.

Thus, the processing unit generates four different kinds of output:

- A1: Information on the synthesized, interpreted situation, passed on as input to the next higher layer ($i + 1$), if present
- A2: Set of instructions that is passed down to the next lower layer ($i - 1$), if present, or that is addressed directly to the technical devices, if $i = 0$
- A3: Newly gathered knowledge rules for the knowledge database
- A4: Information on all inputs, processing steps and

generated outputs, as well as on all modifications that were induced in the parameters of the neural network, to be documented in the logbook

Within the processing unit, algorithmic logic and neural networks can be combined in many different topologies to build the processing unit. In particular, they can be connected in series or in parallel, or in a combination of both, involving one or more instances of both algorithmic logic and neural network.

For example, a modern thermometer, which includes both a temperature sensor and some form of algorithmic logic, could

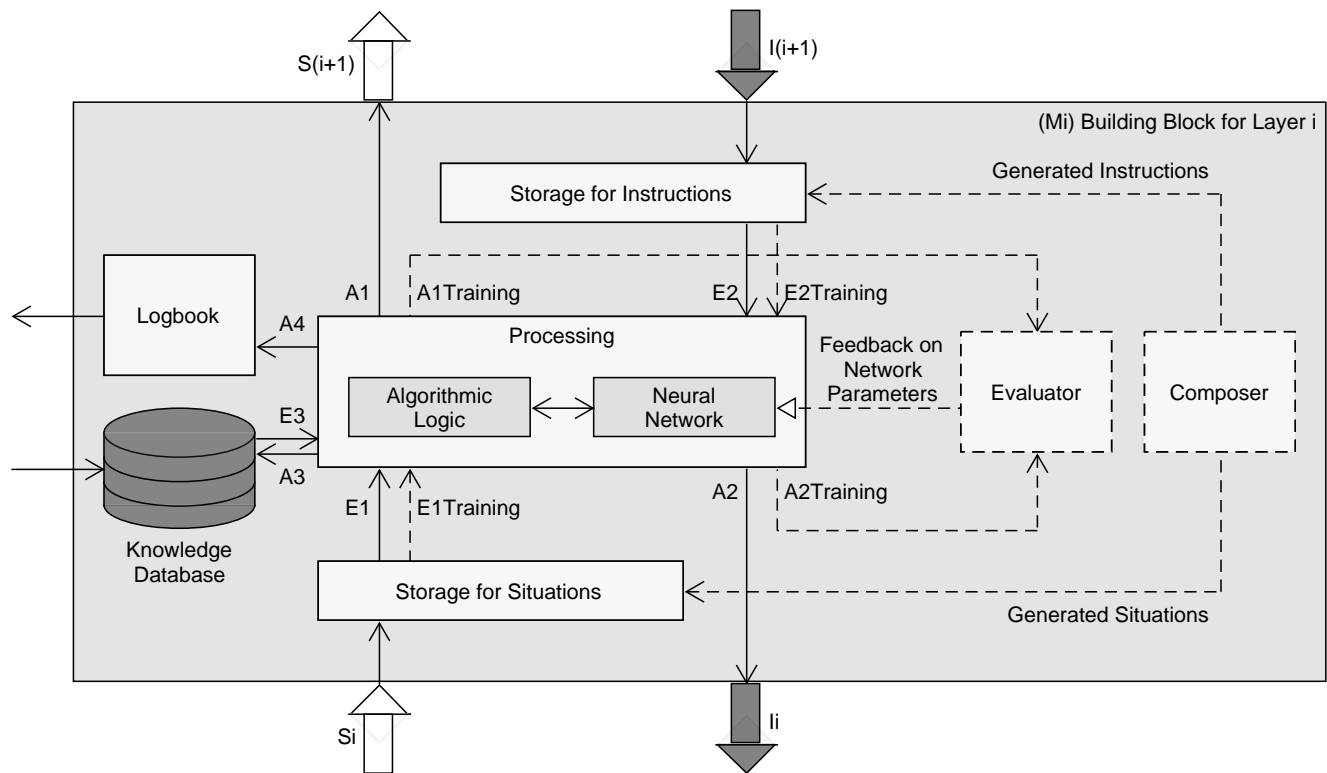


Figure 2. Pattern of building block that implements each layer.

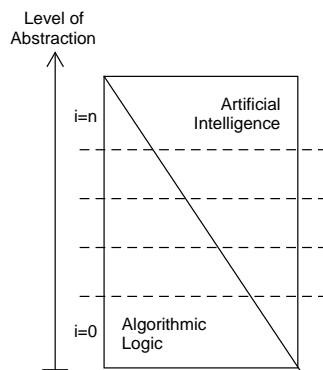


Figure 3. Varying proportions of algorithmic logic and artificial intelligence.

monitor whether the current temperature exceeds a previously defined threshold. If the threshold value is exceeded, the algorithmic logic passes the history of measured values on to the neural network, which synthesizes and analyses this data in order to identify the current situation.

Another example for a neural network connected in-series to a subsequent algorithmic logic (i.e., the other way round from the above example), would be to enter a variety of measured data from different devices into the neural network, which derives from this data the overall situation of the work

place. After the neural network classified and identified the situation, this information on the situation is passed on to an algorithmic logic that executes the predefined process that deals with this type of situation.

An example for running algorithmic logic and neural network in parallel, followed by a second algorithmic logic that is connected in series, would be some sort of security mechanism, where both algorithmic logic and neural network process the same input data individually and independently of each other. After both components reached their classification result, a subsequent algorithmic logic compares the individual results and decides on further processing steps.

Note that this combination of neural network and algorithmic logic constitutes an ensemble. Generally, an ensemble consists of a set of classifiers of different types (e. g. neural network, decision tree, ...) that are trained individually. In any given situation, each classifier computes its own individual prediction. The predictive results that are delivered by the different classifiers are then combined into the prediction of the entire ensemble. More generally, by ensemble methods, we denote meta algorithms that combine several different machine learning techniques into a single predictive model, in order to reduce distortion and to improve the predictive quality. Studies show that the predictive quality achieved by an ensemble is usually more precise than each of the separate classifiers within the ensemble [11].

For training purposes, the building block provides another two components: a composer and an evaluator, depicted in Figure 2 by dashed lines. The composer generates instructions

and situations and inserts them into the respective storages as training data. To achieve this, the composer can either generate this new data from scratch, or cut and paste snippets of “real” data from the storages into new sequences.

During training mode, the processing unit works on this training data *E1Training* and *E2Training* and processes it into the resulting answers *A1Training* and *A2Training*, which are then passed on to the evaluator. The evaluator’s resulting verdict is re-entered into the neural network, to adapt the parameters within the neural network, as necessary. The criteria that form the basis for the evaluator’s assessment are specified by a set of rules, reflecting goals and basic values. For example, they can postulate the maximization of security, or the minimization of costs.

Note that composer and evaluator are active only during training of the neural network, but not during operations.

VI. GENERATING TRAINING DATA THAT DESCRIBES NEW SITUATIONS

By generative adversarial networks (GANs), we denote a class of algorithms in artificial intelligence that is used for unsupervised learning. First ideas on this type of contradictory architectural patterns were developed by [12]. To achieve this, two separate neural networks are needed, which interact by exchanging data. Basically, the two neural networks involved in this approach compete in a zero-sum game [13], where one network acts as the generating model and the other network as a discriminating model [14]. While the second network, i.e., the discriminator, learns how to avoid results that are classified as *bad*, the first network (the generator) tries to challenge the second network so that it delivers a *bad* answer. Thus, over time, the evaluation of the discriminator improves step by step.

Later on, this initial idea of Li, Gauci and Gross was labeled as *Turing Learning* and applied, among others, for generating quasi-realistic photographs [15] [16] [17]. We now transfer the underlying idea into a new context: generating new types of situations that were hitherto unknown, but nevertheless make sense with respect to the laws of physics.

To achieve this, one neural network (the composer, a DCGAN) generates candidates for new situations, which are then evaluated by the other network (the evaluator, a CNN). The evaluator network is trained by presenting it with specific instances of the generated data, until it reaches a satisfactory degree of accuracy [18] when trying to identify the generated data instances from the original real ones. To achieve this, the evaluator calculates an error between the true original and the generated data.

Both networks apply backpropagation to improve their results. As a consequence, step by step the composer learns to create situations that are more and more realistic (and adhere to the laws of physics), while the evaluator improves its ability to correctly identify situations that were synthetically generated.

Once a sufficient accuracy is reached, a thus generated situation may be inserted into the processing unit of the building blocks as regular training data. Then, the composer is again seeded with a new input from the initial data space. On this new, hitherto unknown situation composer and evaluator are trained again. Thus, synthetic situations can be generated over and over again, until no further new situations appear.

Note that it is crucial that situations that were synthetically generated are not entered into the composer/evaluator-pair as original true data. Otherwise, the entire system is in danger of losing its touch of reality. However, it might be interesting to investigate whether in this case, the system would converge towards a single stable state, or whether the latent space shows different plateaus.

In some architectures, the evaluators directly influence the processes that synthetically generate the situation data. An analysis of the consequences of this kind of interdependency between composer and evaluator has yet to be analyzed as part of further research.

VII. EXAMPLE EXECUTION

To illustrate which kinds of tasks are dealt with on the different layers and what kind of information is processed, Figure 4 visualizes the information flow over time for an example system and a specific exemplary situation.

The physical layer of our example work place contains seven devices, six of which are directly connected to the cognitive system: a thermometer *D1*, three cameras *D2*, *D3* and *D4*, a loudspeaker *D5* and an e-mail system *D6*. In addition, the work place comprises a traditional heater *D7*, which is not data connected to the cognitive system, but observed by thermometer *D1* and the three cameras.

Adhering to the generalized architecture in Figure 1, our exemplary cognitive system is structured into four processing layers: *M1* for signal processing close to the technical devices, *M2* for reactions which combines short signal snippets into larger situation contexts, *M3* for drawing conclusions and *M4* for overall organization. Note that layer *M4* merges several work places (*WP2* and *WP3*), in addition to the work place in focus.

In the diagram, time is discretized into time steps, progressing from top to bottom for reasons of readability. (As a consequence, processing layers are arranged vertically, with abstraction increasing from left to right.) Within each time step, all processing actions are executed in parallel. Note that the diagram in Figure 4 abstracts from the processing time that is required in each processing layer. Consider processing to take place at the transition from each time step to its successor, in parallel for each processing layer.

In time step 1, the technical devices pass the data they observed on to layer *M1* for signal processing, as situation information. More precisely, thermometer *D1* communicates a series of measured temperature values, each labeled with a time stamp. All three cameras continuously gather images from their respective sections of the work place. Each camera contains basic image processing facilities, which allow for identification of previously registered work place personnel. Thus, cameras *D2* and *D3* communicate that they identified person “Klaus” at a certain position in the work place. In contrast to this, camera *D4* did not identify any persons in the section of the work place that it observes.

Layer *M1* receives this situation information from the technical devices and stores it in its storage for situations. On this basis, it aggregates the gathered information and synthesizes it into a more complex understanding and larger context of a situation, thus increasing the level of abstraction. Here, layer *M1* realizes that both cameras *D2* and *D3* identified the same

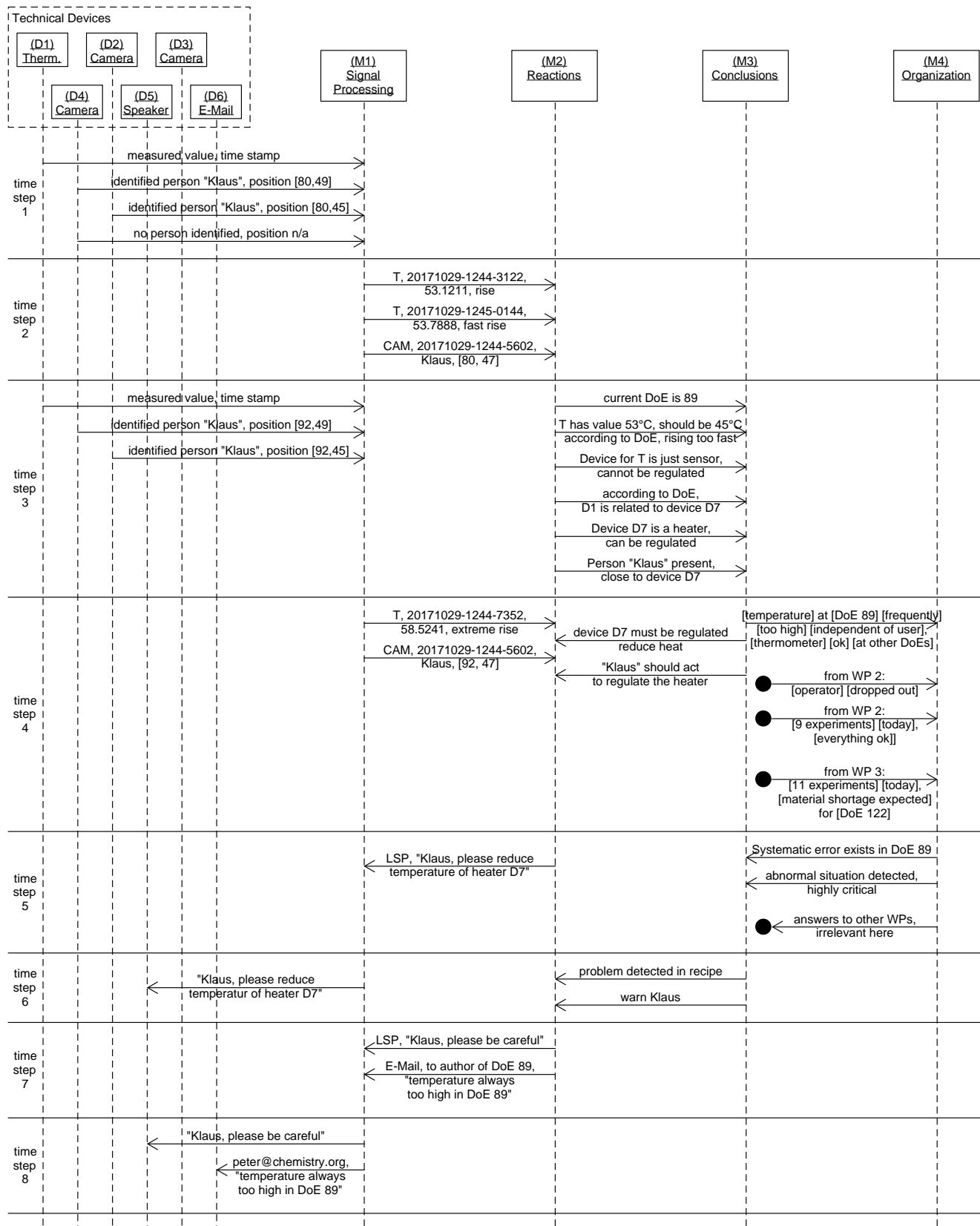


Figure 4. Information flows between processing layers, for an exemplary execution of the system.

person, Klaus, and calculates the position of Klaus in the work place. Furthermore, layer *M1* analyzes the sequence of temperature information measured by the thermometer. Here, layer *M1* realizes that the temperature rises really quickly. This aggregated information is then passed on to layer *M2* for reactions in time step 2. Information is organized according to the syntactic pattern of type of device, time stamp and two more items of structured information, whose syntax and semantics are relative to the type of the device.

The knowledge database of layer *M2* contains information on the experiments that are carried out within the work place, referenced as DoE (i.e., design of experiments) in Figure 4. From previous context information, layer *M2* is aware that DoE 89 is currently processed. *M2* realizes that the measured temperature rises both faster and higher than specified in the “recipe” that is defined in DoE 89, and that the thermometer *T* is correlated with the heater *D7*. As well, *M2* identifies that the thermometer *T* is merely a sensor and thus cannot be regulated, whereas the heater *D7* can be regulated. Furthermore, *M2* identifies that person Klaus is located close to the heater *D7*. All this synthesized situation information is passed on to layer *M3* for conclusions in time step 3, and stored there in *M3*’s storage for situations. In addition, the technical devices keep sending situation information towards level *M1* continuously. In Figure 4, this information flow is indicated as well for time step 3.

As a next step, layer *M3* deduces from the situation information that device *D7* is about to overheat and that Klaus is still present and able to act. Furthermore, experience gathered from previous situations indicates that in experiments that are executed according to DoE 89, temperature problems arise rather frequently, independently of the current human operator. In addition, *M3* realizes that heater and thermometer work fine in other experiments, and thus seem to be in order technically. As a result, in time step 4 layer *M3* communicates as instructions to level *M2* that the heater *D7* must be regulated, and that Klaus should act to regulate the heater. In addition, *M3* communicates to level *M4* for organization that temperature problems occur frequently when executing DoE 89. In addition, during time step 4 layer *M1* passes on its newly aggregated situation information on to level *M2*, indicating that the temperature is still rising and that Klaus has moved towards the heater. As layer *M4* for organization joins the information of several work places, additional information arrives as input from other work places during time step 4.

Based on all this situation information, layer *M4* for organization deduces that there might exist a systematic problem in the documentation of DoE 89, such as wrong instructions. As well, *M4* identifies that the situation in work place 1 is highly critical. Therefore, *M4* specifies suitable instructions and passes them down to level *M3* during time step 5. In parallel, layer *M2* processes the newly arrived situation information in the light of the instructions that were handed down towards layer *M2* during time step 4. As the temperature is still rising rapidly, *M2* passes down to layer *M1* the instruction that the loud speaker *D5* should instruct Klaus to reduce the temperature of heater *D7*.

This instruction is translated by layer *M1* into appropriate signals for the loud speaker *D5*, which are transferred to *D5* during time step 6. In addition, layer *M3* for conclusions derives from the instructions received from *M4*, in combination

with the information on the ongoing situation, that a problem was detected in the recipe of DoE 89 and that Klaus has to be warned about the critical situation. Corresponding instructions are passed from *M3* to *M2* during time step 6.

M2 translates these abstract instructions into device specific instructions and passes them down to signal processing *M1* during time step 7.

Finally, *M1* generates the appropriate, device specific signals for the loud speaker *D5* and for the e-mail system *D6*, respectively, and passes them down to their respective recipients, which execute them appropriately.

VIII. PROTOTYPICAL PROOF OF CONCEPT

A prototypical proof of concept addressing the lower layers of the example presented here was realized as a show case, using IBM Watson [19] [20] as well as Tensorflow [21] [22] for the neural network processing.

In this prototypical realization, the layer *M0* comprises a variety of cognitive channels implemented via IoT hardware: three cameras, one projector, one microphone, speakers addressed via five Raspberry PI (based on Python), one pH-Meter, one scale, and temperature sensors addressed via three ESP8266 (based on Lua). A selection of IBM Watson services is used to realize cognitive abilities on this layer (STT speech to text, TTS text to speech, and visual recognition via REST).

Both layers *M1* and *M2* are realized in the cloud. Processing is based on IBM Bluemix Services [23] that are implemented in TypeScript. In addition, more complex cognitive abilities from the IBM Watson portfolio are included on layer *M2*, e.g., NLC natural language classifier, which is addressed via REST as well. Furthermore, the storage for situations on layer *M2* is realized via a NoSQL Couch database.

Layer *M3* runs on local hardware. Processing logic is implemented in Python. Situation information is retrieved from *M2* by download. Within the prototype, test cases were classified by hand and are processed by a convolutional neural network (based on MNIST implementation) in Tensorflow, using two convolutional / pooling layers and the dense layer with ten cases. Classification results are uploaded manually to layer *M2*.

In spite of the rather small number of test cases, the system achieves good classification results. Note that for layer *M3* to deliver more comprehensive classification results, a much larger amount of data would have to be collected on level *M2*, comprising at least 1000 laboratory days. Nonetheless, by this prototypical realization and the test cases under consideration, it was possible to validate the feasibility of our architecture.

Note that for *M4*, an even greater amount of data would be necessary, due to the complexity of decisions and reasoning that take place in this layer. As there are no predefined networks available for this domain, training has to be executed from scratch. Thus a large amount of “experience” in terms of data must be gathered before more meaningful results can be achieved on this layer.

IX. CRITICAL DISCUSSION

In principle, it would be possible to implement a cognitive assistant system with matching abilities as a monolithic block of neural networks, rather than using our multilayer architecture. However, this monolithic block would have to handle the

entire complexity by itself, thus requiring an extreme amount of training that greatly exceeds what can be handled even by modern hardware. This complexity can be handled only by partitioning the system into smaller parts that are implemented and trained separately.

All in all, the structure of the building blocks ensures that the system corresponds to a sequence of symbolic components (for persistently storing information on situations and instructions) and subsymbolic, algorithmic components that process this information. Thus, the overall processing is clearly structured into distinct layers, which facilitates the implementation of processing units and allows for their individual training, as well as for the analysis of the resulting information.

Note that the layering we suggest does *not* replicate the layers of the human brain. Rather, it creates levels of abstraction that are tailored to meet the specific requirements of the technical system. As a consequence, the system's cognition, and thus, its awareness, will differ from that of a human being.

As any artificial intelligence, the system has an error margin that depends, e.g., on the noisy environment, changing illumination, the possible novelty of input sequences, as well as on the imperfection of the decision system itself. Thus, it is possible that the system misinterprets a situation.

If, for example, the system's task is to identify a person "Klaus" based on data gathered by cameras and microphones, it can indeed happen that Klaus is not recognized, or that a wrong person is recognized as "Klaus" (although it is, in fact, "Peter"). In the first case, the system is aware that something did not work properly; in the second case, it is not.

Strategies for dealing with the first error case range from *retry* (i.e., issuing instructions to present oneself to the camera again) to *comment*, thus informing the user as comprehensively as necessary that something unexpected has happened. The second case, where the system is unaware of its error, is more severe. Although it cannot be entirely avoided, its possibility can be significantly reduced by sufficient training.

Currently, the person identification process is based on the matching of people's looks, i.e., on rather static optical features, such as the shape of the face or the hair colour. To improve security, voice recognition could be added. In addition, it would be possible to analyse and compare typical behavioral patterns of human actors, such as their specific way to execute certain movements, e.g., the way they walk or their body pose while working. As behavioural patterns are much harder to copy than mere static looks, this could increase security. However, to be able to differentiate small nuances in movements in order to correctly identify individual people, a vast amount of additional data and training would be required.

Another possibility for increasing security would be for the system to compare the current interaction pattern of the person identified as "Klaus", who is working right now in the laboratory, with history data on previous interactions between Klaus and the system. If the system realizes that Klaus acts and reacts differently than usual, e.g., shows different response times, or executes the different steps in a faster or slower way, it could issue a warning to some other control unit (human or otherwise), indicating the possibility that the person in the laboratory might not be "Klaus" after all; or that it is indeed Klaus, but Klaus on a bad day (i.e., headachy or preoccupied), and thus not acting up to his usual well-focused, competent

self. Both cases would require some action to reinstate the overall system's security.

Note that layer $M2$, which is responsible for realizing rather basic reactions, incorporates several algorithmic mechanisms for identifying, and then blocking, discrepancies to usual patterns, contradictions to what is expected and desired, maloperation or even blatant misuse. In contrast to this, incorrect decisions on layer $M4$, in our example responsible for the overall organization, are much harder to tackle. Similar to decision processes in biological systems, they can only be tackled by increased training and subsequent debating [24].

In each building block, processing is performed by a combination of neural networks and algorithmic logic, which are integrated into an ensemble. Note that the proportions of the different parts varies, depending on the level of abstraction of the layer that is realized by the building block. Thus, each layer can incorporate different components that are self-contained and pre-trained individually. Examples for these components are a device for identifying laboratory glassware, a security check or a process control logic.

We expect that in future versions of our system, each processing unit will be manually composed from a set of individual modules, neural networks or algorithms, as suggested by [25] [26]. This development is supported by the increasing possibilities for acquiring modules that are pre-trained for certain tasks. Platforms such as Model Asset eXchange (MAX) from IBM or algorithmia facilitate the trading with pre-trained models.

X. CONCLUSION AND FUTURE WORK

We introduced a multilayer architecture for cognitive systems that support the operation of technical work places, in which hybrid processes (partially executed manually, and partially using technical devices) are executed. To ensure efficiency and adaptability, we structured this architecture into separate layers on different levels of abstraction. Each layer deals with specific kinds of tasks and processes the corresponding kind of information, which again is organized into different levels of abstraction.

Each layer of the conceptual architecture is realized by a building block, which incorporates aspects of both algorithmic logic and artificial intelligence. We provided a template defining the glass box view of these building blocks. Based on this template and the conceptual architecture, it is possible to develop cognitive systems that scale appropriately, to meet the demands of the application context under consideration.

To ensure adequate training of the different processing units, each building block incorporates generative adversarial networks that cooperate as composer and evaluator, in order to generate training data that exceeds situations that have been physically measured in the past.

As a next step, we demonstrated the interaction and cooperation of the different layers for a concrete example, specified from the context of a chemical laboratory. Furthermore, we sketched a prototypical proof of concept that addresses the lower layers of the presented example. This prototype was run on a small number of test cases, to validate the feasibility of our architecture.

For extending the prototypical system towards a more comprehensive classification of situations and recommendations of

instructions, extensive laboratory data will have to be collected, as a basis for properly training the cognitive system.

Currently, introducing an assistant system that is based on the architecture presented here, into the workplaces of a large German manufacturer of spectacles and glasses is under discussion [27]. Generally, our system architecture is best suited for supporting workplaces of a highly technical and scientific character, such as the premises of optometrists or hearing aid acousticians, as well as for laboratory environments in a chemical, pharmaceutical or medicinal context.

Parts of this work are closely related to an innovation that is covered by the German patent application 10 2017 126 457.4.

REFERENCES

- [1] V. Thurner and T. Gressling, "A Multilayer Architecture for Cognitive Systems – Supporting well-defined processes that are partially executed manually in technical work places," in *Cognitive 2018 – The Tenth International Conference on Advanced Cognitive Technologies and Applications*, Barcelona, Spain. IARIA, 2018, pp. 63–71.
- [2] M. Burgin and G. Dodig-Crnkovic, "A Taxonomy of Computation and Information Architecture," in *ECSAW*, Dubrovnik, Croatia. ACM, 2015, pp. 7:1–7:8, DOI: 10.1145/2797433.2797440.
- [3] C. Leibold and M. Spies, "Towards a Pattern Language for Cognitive Systems Integration," in *EuroPLoP*, Irsee, Germany. ACM, 2014, pp. 17:1–17:9, DOI: 10.1145/2721956.2721968.
- [4] M. Aehnelt and B. Urban, "The knowledge gap: Providing situation-aware information assistance on the shop floor," in *HCI*, Los Angeles, USA. Springer, 2015, pp. 232–243, DOI: 10.1007/978-3-319-20895-4_22.
- [5] O. Korn, M. Funk, and A. Schmidt, *Assistive systems for the workplace*. IGI Global, 2015, pp. 121–135, DOI: 10.4018/978-1-4666-8200-9.ch097.
- [6] T. Kosch, Y. Abdelrahman, M. Funk, and A. Schmidt, "One size does not fit all – Challenges of providing interactive worker assistance in industrial settings," in *UbiComp/ISWC*, Maui, USA. ACM, 2017, pp. 1006–1011, DOI: 10.1145/3123024.3124395.
- [7] A. Srivastava and P. Yammyiyavar, "Design of multimodal instructional tutoring agents using augmented reality and smart learning objects," in *ICMI*, Tokyo, Japan. ACM, 2016, pp. 421–422, DOI: 10.1145/2993148.2998531.
- [8] M. Funk, T. Kosch, and A. Schmidt, "Interactive worker assistance: Comparing the effects of in-situ projection, head-mounted displays, tablet, and paper instructions," in *UbiComp*. ACM, 2016, pp. 934–939, DOI: 10.1145/2971648.2971706.
- [9] S. Büttner, O. Sand, and C. Rucker, "Exploring design opportunities for intelligent worker assistance: a new approach using projection-based AR and a novel hand-tracking algorithm," in *AmI*, Malaga, Spain. Springer, 2017, pp. 33–45, DOI: 10.1007/978-3-319-56997-0_3.
- [10] J. Anderson, M. Matessa, and C. Lebiere, "ACT-R: A Theory of Higher Level Cognition and its Relation to Visual Attention," *Human-Computer Interaction*, 1997.
- [11] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling Neural Networks: Many could be better than All," *Artificial Intelligence*, vol. 137, 2002, pp. 239–263.
- [12] J. Schmidhuber, "Learning Factorial Codes by Predictability Minimization," *Neural Computation*, vol. 4, no. 6, 1992, pp. 863–879.
- [13] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and J. Bengio, "Generative Adversarial Networks," *arXiv* 1406.2661, 2014.
- [14] W. Li, M. Gauci, and R. Gross, "A Coevolutionary Approach to Learn Animal Behavior Through Controlled Interaction," in *GECCO*, Amsterdam, Netherlands, 2013, pp. 223–230.
- [15] C. e. a. Ledig, "Photo-Realistic Single Image Super-Resolution Using a Generative Adversarial Network," *arXiv [cs.CV]*, 2016.
- [16] S. e. a. Reed, "Generative Adversarial Text to Image Synthesis," *arXiv [cs.NE]*, 2016.
- [17] H. e. a. Zhang, "Text to Photo-Realistic Image Synthesis with Stacked Generative Adversarial Networks," *arXiv [cs.CV]*, 2016.
- [18] T. Karras, T. Aila, S. Laine, and J. Lehtinen, "Progressive Growing of GANs for Improved Quality, Stability, and Variation," *arXiv [cs.NE]*, 2017.
- [19] K. Mak, H. Pilles, M. Bertl, and J. Klerx, "Wissensentwicklung mit IBM Watson in der Zentralkodokumentation (ZentDok) der Landesverteidigungsakademie," https://www.academia.edu/37674334/Wissensentwicklung_mit_IBM_Watson, accessed 11/2018, Landesverteidigungsakademie, Tech. Rep.
- [20] Y. Saxena, "IBM Watson: Determining the presence of an Artificially Intelligent Nature," https://www.academia.edu/23441941/IBM_Watson_Determining_the_presence_of_an_Artificially_Intelligent_Nature, accessed 11/2018.
- [21] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, "Tensorflow: A system for large-scale machine learning," in *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*. Savannah, GA: USENIX Association, 2016, pp. 265–283. [Online]. Available: <https://www.usenix.org/conference/osdi16/technical-sessions/presentation/abadi>
- [22] F. Ertam and G. Aydın, "Data classification with deep learning using Tensorflow," in *2017 International Conference on Computer Science and Engineering (UBMK)*, Oct 2017, pp. 755–758.
- [23] J. Moore, M. Hirzalla, R. Osowski, S. Chowdhury, and V. Gucer, "IBM Bluemix Architecture Series: Web Application Hosting on Java Liberty – Leveraging best practice and reference architectures for cloud," <http://www.redbooks.ibm.com/redpapers/pdfs/redp5184.pdf>, accessed 11/2018, IBM, Tech. Rep.
- [24] IBM Research, "IBM Project Debater – Can artificial intelligence expand a human mind?" <https://www.research.ibm.com/artificial-intelligence/project-debater/>, accessed 11/2018.
- [25] C. Shu and D. H. Burn, "Artificial Neural Network Ensembles and Their Application in Pooled Flood Frequency Analysis," *Water Resources Research*, vol. 40, 2004, p. 655.
- [26] X. Yao and M. M. Islam, "Evolving Artificial Neural Network Ensembles," *IEEE Comput. Intell. Mag.*, vol. 3, 2008, pp. 31–42.
- [27] ARS Computer und Consulting GmbH, "ARS Beratung – Architektur und Cognitive Services – Rodenstock GmbH," https://web.ars.de/wp-content/uploads/2018/04/ARS_Referenz_Rodenstock_Machbarkeitsstudie_Cognitive_AI.pdf, accessed 11/2018.

Novel Field Oriented Patient Monitoring Platform for Healthcare Facilities

Yoshitoshi Murata, Rintaro Takahashi, Tomoki Yamato

Faculty of Software and Information Science
Iwate Prefectural University
Takizawa, Japan

e-mail: y-murata@iwate-pu.ac.jp, {g031n100, g031n161}@s.iwate-pu.ac.jp

Shohei Yoshida, Masahiko Okamura

Graduate School of Software and Information Science
Iwate Prefectural University Graduate School
Takizawa, Japan

e-mail: {g231n034, g231p005}@s.iwate-pu.ac.jp

Abstract— Several kinds of patient monitoring systems have been provided to healthcare facilities such as hospitals in recent years. Most of these systems lack interoperability and are designed for sensors to be connectable within the same medical manufacturer. In these systems, monitoring devices and terminals are connected to a network, and operations are performed on a central console. This means that an operator is needed to do the job, which is not suitable for realistic healthcare fields. To address these problems, we have developed and propose a novel patient monitoring platform for healthcare facilities. This platform consists of a notification system for event monitoring applications and a data collecting system for data measuring applications. In the notification system, pairing between a patient monitoring device and a mobile terminal carried by a medical person can be completed merely by reading a Quick Response Code on the display of a monitoring terminal at a patient's bedside. When the monitoring device detects that a patient is having trouble, it sends event messages to the mobile terminal. In the data collecting system, medical data files measured by sensor devices are automatically uploaded to a cloud server and downloaded from the server to an authorized medical professional.

Keywords—hospital; patient monitoring system; remote measuring system; patient-nurse hotline; QR-code; healthcare facility.

I. INTRODUCTION

We are developing patient monitoring systems [1]. Several kinds of communication systems have been provided to medical or healthcare facilities in recent years. They are roughly classified into call systems and patient monitoring systems. The latter are roughly classified into event monitoring systems and data measuring systems, which are sometimes integrated as a comprehensive network system.

In event monitoring applications, notifications that a patient is having problems are usually sent to a nurse station or management office rather than to a specified medical person in Japan. Therefore, the nurse station or office is likely to be very busy quite frequently. An example of the types of systems used in Japan is the patient-nurse hotline system provided by Carecom Inc. [2]. In this system, several kinds of sensor devices are connected to the hotline. One such device is a mat sensor to detect a patient leaving his/her

bed [3]. The mat is beside the bed, and if a patient steps on it, an alert is sent to a nurse station.

Honeywell provides a tracking and localization system integrated with a patient communication system and a call system [4]. General Electric Company (GE) provides many kinds of patient monitoring equipment [5]. They are connected to a central computer server through a hospital intranet. This makes it possible for medical professionals to monitor measured data.

Most of these systems lack interoperability and are designed for sensors to be connectable within the same medical manufacturer. In these systems, monitoring devices and terminals are connected to a network, and operations are performed on a central console. This means that an operator is needed to do the job, which is not suitable for realistic healthcare fields.

To address the problems of the lack of interoperability, necessary of an operator and busy nurse stations or offices, we developed a patient monitoring platform designed for healthcare fields. This platform consists of a notification system for event monitoring applications and a data collecting system for data measuring applications.

We first proposed the notification system at eTELEMED 2018 [1]. In this system, the monitoring device by the patient's bedside is connected to a mobile terminal such as a smartphone carried by a medical worker. The mobile terminal reads a Quick Response Code (QR-code) [6] on the monitoring device to pair them. This operation can be done at the bedside of a patient. We assume that it would be done by a medical worker such as a nurse who establishes a monitoring device for the patient. When the monitoring device detects the patient is having trouble, it sends a notification message to the mobile terminal. The medical worker who receives the message immediately goes to the monitored patient. The sensor relay unit and the mobile terminal were developed with an Android smartphone. The sensor relay unit is a part of the monitoring device to which sensors are connected. This makes it possible to hand the patient monitoring operation to another worker merely by having the latter read the QR-code on their smartphone. This operation is the same as reading the original QR-code.

In addition to the notification system, we also developed a novel data collecting system for measuring applications. In this system, measured data are transferred to a cloud server and stored on it. Authorized persons such as medical

professionals can access the server to download and analyze the stored data. A supervisor can freely change the relationship established among patients, medical workers who provide measuring terminals and medical professionals who have access to servers.

We developed two event monitoring applications that use the proposed notification system and one data measuring application that uses the data collecting system. The first one monitors cases when intravenous feeding devices are removed from a patient, the second one monitors cases when the patient leaves the bed, and the third one continuously measures changes in the angle of the arm and the lumbar region of the body.

After introducing related work in Section II, we describe the notification system and its example applications in Section III. The data collecting system and example applications for it are detailed in Section IV. Section V concludes with a summary of key points.

II. RELATED WORK

Remote monitoring applications for patients are roughly classified into event monitoring applications and measuring applications. In this section, both of them are introduced.

A. Event monitoring applications

These applications have been put to use ever since communication systems for sending messages from monitoring devices to hospital personnel first started to be used.

One example is the “Risho Catch” system Paramount Bed Inc. has developed [7]. “Risho” means getting out of a bed in Japanese. This system detects when a patient sits up in bed, sits on the side of the bed, or leaves the bed. When this happens, it sends a message to a nurse station through a patient-nurse hotline system. The system structure is shown in Fig. 1. A load sensor unit in the bed is connected to the patient-nurse hotline system through a relay unit located at the patient’s bedside. The nurse station has a monitor and console terminal. It is possible to send messages to a mobile terminal. However, wireless tablets and smartphones are not commonly used for sending messages to nurse stations in Japan. Therefore, nurse stations and offices are likely to often become very busy.

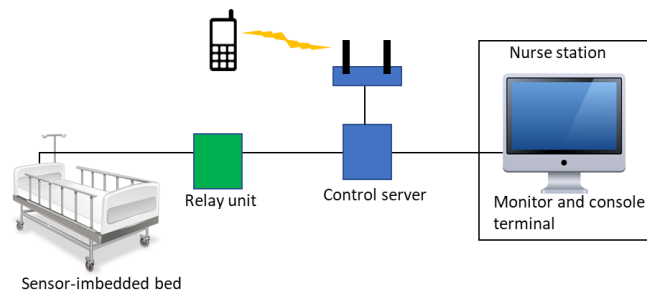


Figure 1. System and network structure of “Risho Catch.”

Balaguera et al. evaluated decreasing the number of falls from bed using the SensableCare System [8]. Its architecture

is shown in Fig. 2. The sensor pad in the system sends data through a cable to the control box located at the patient’s bedside. The control box wirelessly transmits this data to a Bluetooth access point located throughout the ward. This information then travels through the hospital WiFi network to the dashboard and docking server where the data is analyzed. When an alert is sent to the nurse via an application on his/her mobile terminal, it is wirelessly transmitted through the hospital WiFi network. The patient’s condition is monitored on the dashboard terminal.

In both systems, the console terminal in the nurse station or office must connect sensor units with a mobile terminal. This makes it a little hard to change relationship between a mobile terminal and monitoring patients.

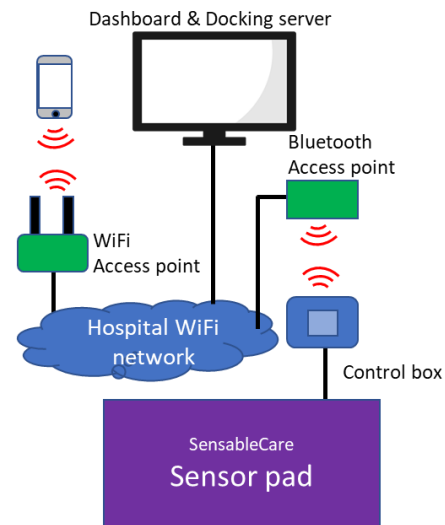


Figure 2. Architecture of the SensableCare System.

B. Measuring applications

In the early 2000s, there were several remote measuring systems for patients that used the General Packet Radio Service/Wireless Local Area Network (GPRS/WLAN) as the wireless network and the Personal Digital Assistance (PDA) as the mobile device [9-12]. In these systems, sensor devices were connected to a monitoring terminal or server through a wireless network. Since they were experimental systems, they had no fixed destination address.

In recent years, several companies have been providing not only patient remote monitoring devices but also cloud services. GE provides a “GE Health Cloud” system along with many kinds of sensor devices and monitor devices [13]. The cloud manages the connecting of sensor devices to the hospital network and operates them on the console terminal.

This scheme maintains a high security level but lacks flexibility and interoperability. This makes it difficult for medical workers to install and pair sensor devices at the patient’s bedside, especially other company’s devices.

III. NOTIFICATION SYSTEM

In this section, we describe the novel notification system for event monitoring applications.

A. Design concept

We designed the system so that:

- (1) Medical personnel can install monitoring devices.
- (2) Medical personnel can easily pair the monitoring devices with their own mobile terminals at the patient's bedside.
- (3) When the monitoring device detects a patient is having problems, it sends notification messages to the mobile terminals of the medical personnel, not a nurse station.
- (4) Pairing situations can be monitored from a console terminal.
- (5) Event messages are monitored from a console terminal.
- (6) Mobile terminals belonging to other organizations must be excluded.

Therefore, no operations are performed with the console terminal and monitoring devices can be easily installed at the patient's bedside. The other side, the supervisor such as a medical doctor in charge, can monitor pairing situations and event messages to maintain security and safety. The system configuration is shown in Fig. 3. In this figure, Worker A monitors a patient Mr. P at first and hands off the monitoring job to Worker B later. There is no alarms in a nurse station.

The system consists of three programs; one works on the sensor relay unit to which sensor units are connected, the second one works on mobile terminals to receive event messages and present them, and the third one works on the cloud server to relay event messages from the sensor relay unit to the mobile terminals.

Prior to monitoring a patient, each terminal must register with the cloud server and get an ID from it. Here, we call the ID for the sensor relay unit "SR-ID", that for Worker A's mobile terminal "MA-ID", and that for Worker B's mobile terminal "MB-ID." When one of the two mobile terminals establishes pairing with the sensor relay unit, it must send SR-ID together with its own ID (MA-ID or MB-ID). A QR-code is generated on the sensor relay unit from SR-ID and presented on its display. Since the QR-code encodes text data to codes and encodes code to text data, using the QR-code enables a worker to enter SR-ID easily. If Worker A pairs his/her mobile terminal with the sensor relay unit, he/she merely reads the QR-code on the sensor relay unit's display with his/her mobile terminal to input SR-ID.

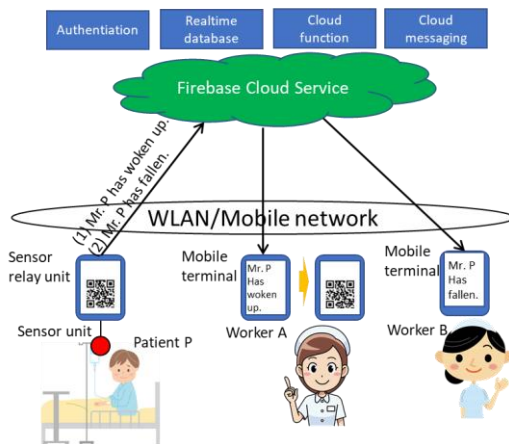


Figure 3. Configuration of the notification system.

The mobile terminal sends its own ID (MA-ID) and the read ID (SR-ID) to the cloud server. The cloud server uses SR-ID and MA-ID to pair the sensor relay unit and Worker A's mobile terminal.

In the case shown in Fig. 3, the sensor unit detects the event "waking up in bed" and the sensor relay unit sends the message "Mr. P has woken up" to Worker A's mobile terminal via the cloud server.

This system using a QR-code is useful for handing monitoring work over to another worker. The handing over operation is done by a worker (Worker B in Fig. 3) reading a QR-code on his/her mobile terminal. This QR-code is created from the SR-ID read from the sensor relay unit. In this situation, event messages can be sent to both mobile terminals. However, no messages will be sent to Worker A's mobile terminal if Worker A sends the signal to release his/her pairing with the cloud server. In the case shown in Fig. 3, the second message "Mr. P has fallen." was sent to Worker B.

B. System configuration

Authentication and messaging functions are needed to meet the requirements listed at the beginning of subsection A. Hence, we decided to use the Google Firebase Cloud Service (GFB) [14] to implement the notification system. The GFB has many functions; the ones we use are an authentication function to exclude non-registered terminals, a real-time database to manage and monitor the status of pairings and event messages, and a push messaging function to send notifications. As mentioned above, our system consists of three programs. These programs we developed are "PatientApp", which works on the sensor relay unit, "NurseApp", which works on the mobile terminals, and "MessageManager", which works on the GFB. We use an Android smartphone for the sensor relay unit and the mobile terminals. We designed PatientApp and NurseApp from a simple application and a corresponding library program so that a practical event monitoring system that adopts kinds of sensors could be developed easily.

The MessageManager program works with the Authentication, Realtime Database, Cloud Function, and Cloud Messaging tools in Firebase. These tools can be summarized as follows:

- Authentication: This provides backend libraries to authenticate users for developing applications. It supports authentication by using passwords, phone numbers, and popular federated identity providers such as Google, Facebook, and Twitter.
- Realtime Database: This is a cloud-hosted database. It stores data in JSON format and synchronizes it in real time to every connected client.
- Cloud Function: This automatically runs backend codes in response to events triggered by Firebase features and HTTPS requests. It stores backend codes in Google's cloud and runs in a managed environment.
- Cloud Messaging: This is a cross-platform messaging solution that delivers messages reliably. It can send notification messages to drive user re-engagement and retention.

We introduce sequence flows between these tools when building and registering a PatientApp or NurseApp program, pairing between PatientApp and NurseApp, and sending notifications. We use the Open Source QR Code Library [15] to create a QR code in the sensor relay unit and the mobile terminals. In the following sequence flows, the program is described as “QR maker.”

1) Building PatientApp and NurseApp (Fig.4)

Before developing a Firebase application, a developer accesses Firebase to download a configuration file. Firebase sends back a configuration file that includes the project code and software development kit as a Google-Service.json file to a development PC.

After building an application, an Android application package file (Apk File) is made as a building application. Each application connects to Firebase with a Project ID and Application ID. These IDs are included in the Google-Service.json file in the building process.

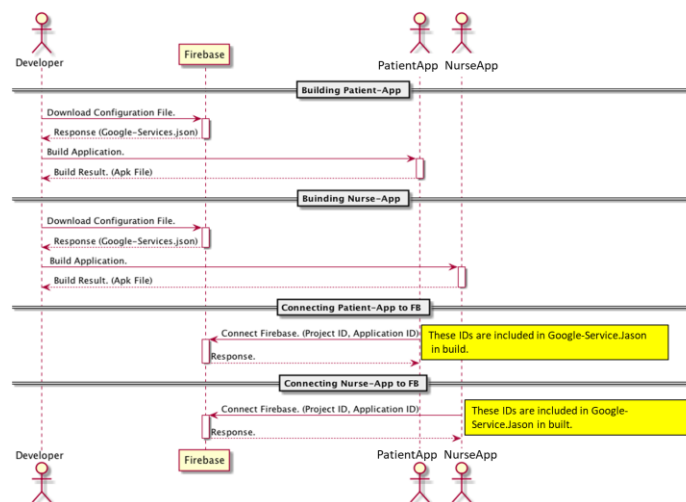


Figure 4. Sequence flow to build applications.

2) Registering PatientApp and NurseApp (Fig. 5, 6)

Since mobile terminals are commonly used by multiple medical workers, a proper login procedure is needed to identify the current worker. We currently use the Google account certification for the NurseApp program to register a medical worker with Firebase. It is possible to change from the Google ID to another identification code. The NurseApp program works with the MessageManager program, which assigns the authentication procedure to the Authentication in Firebase. Then, an account (Nurse UID) is created and managed in the Realtime Database as shown in Fig. 5. The Nurse UID corresponds to the MA-ID or MB-ID given in subsection A.

Since the sensor relay unit itself is registered with MessageManager, we use the Anonymous certification in this case. An account for the PatientApp (Patient UID) is automatically created and managed in the Realtime Database as shown in Fig. 6. The Patient UID corresponds to the SR-ID given in subsection A.

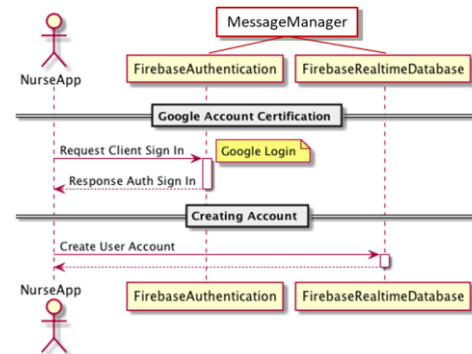


Figure 5. Sequence flow to register a mobile terminal.

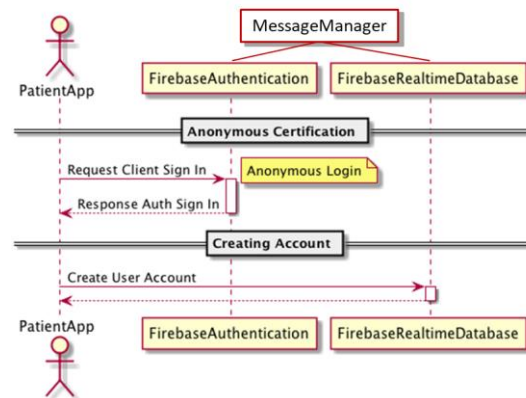


Figure 6. Sequence flow to register a sensor relay unit.

3) Pairing between PatientApp and NurseApp (Fig. 7)

The PatientApp accesses the Authentication to get its own current user (Patient UID, token) and the QR maker makes a QR-code from a Patient UID. NurseApp gets a Patient UID to read the QR code, accesses the Authentication to get its own current user (Nurse UID, token), and accesses the Realtime Database to write the pairing information (token, Patient UID, Nurse UID).

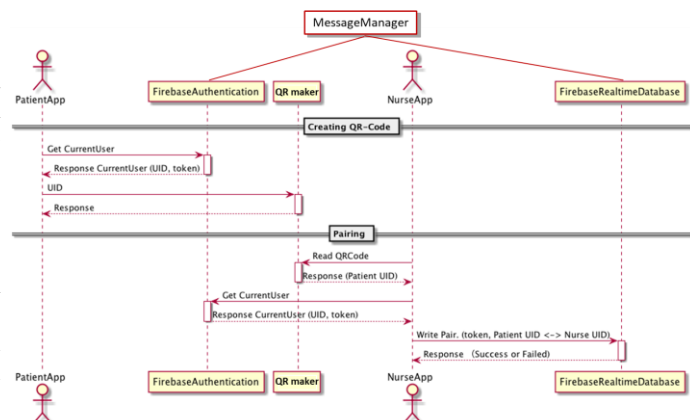


Figure.7 Sequence flow to pair between PatientApp and NurseApp.

4) Sending notifications (Fig. 8)

When a PatientApp in the sensor relay unit detects an event, it accesses the Authentication to get its own current user (Patient UID, token) and the Realtime Database to change its own state.

The Realtime Database collaborates with the Cloud Function to request the Cloud Messaging to push a notification (Nurse push token). The Cloud Messaging sends a signal (Notify Patient State Change) to a NurseApp as shown in Fig. 8.

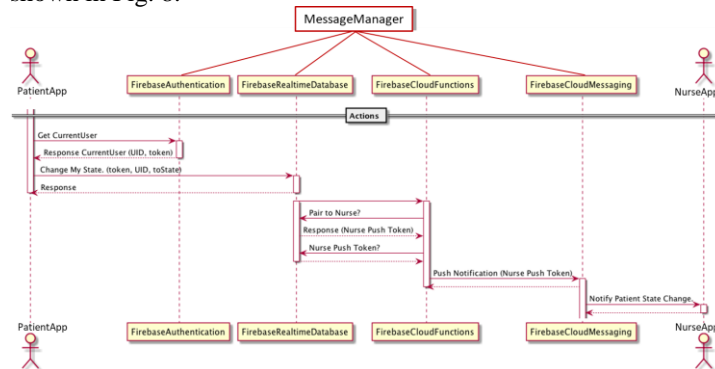


Figure 8. Sequence flow to send an event message.

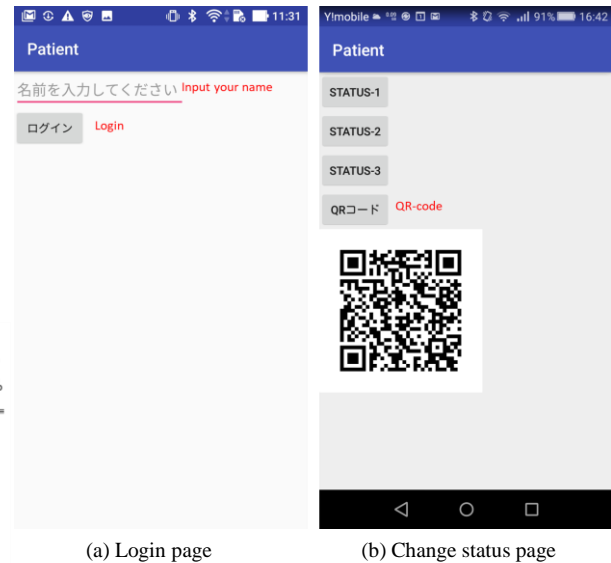
C. Test application and management display

We developed a very simple PatientApp and NurseApp to test the proposed notification function. We call them PatientTest and NurseTest. Since these application programs were developed for Japanese, each item on UI pages was written in Japanese. Hence, we provided an English translation for each item.

After logging in with the Anonymous certification of PatientTest, the four buttons “STATUS-1”, “STATUS-2,” “STATUS-3,” and “QR-code” are presented as shown in Fig. 9. The first three buttons are used to change the status in a sensor device. Clicking the “QR-code” button creates a QR-code from a Patient UID and shows it on the display (Fig. 9 (b)).

After the Gmail address and password are input to the NurseTest login page (Fig. 10 (a)) and the PatientTest QR-code (Fig. 9 (b)) is read, the list page of patients (Fig. 10 (b)) is presented. In the case shown in Fig. 10 (b), the nurse has established two sensor relay units and is monitoring two patients. Patient “Murata test” is having trouble. Clicking the “Messages” button, a list of received messages is presented as shown in Fig. 10 (c). When the “Handling” button is clicked, the same QR-code as that shown in Fig. 9 (b) is presented. Messages and the relationship between sensor devices and monitoring nurses are managed in the Realtime Database. Supervisors are able to monitor them by accessing Firebase.

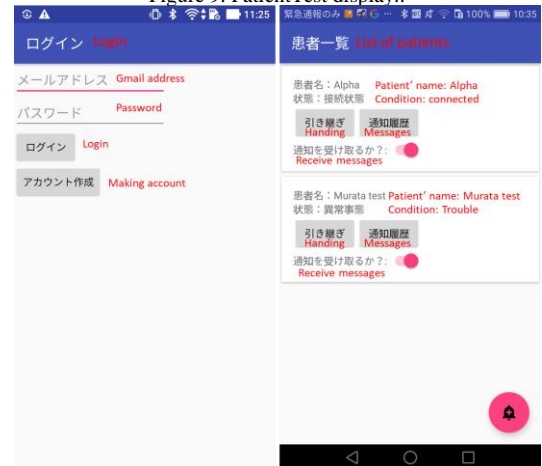
When Firebase is accessed, and the Realtime Database is clicked, the display appears as shown in Fig. 11. Information is classified into “notifications,” “nurses” and “patients.”



(a) Login page

(b) Change status page

Figure 9. PatientTest display..



(a) Login page







(b) List page of patients







(c) List page of messages

Figure 10. NurseTest display.

In Fig. 11 (a), the notifications are classified into those for two nurses whose encrypted Nurse UUIDs are “4ANVMkv1kDarruVlt8l1opmAYj13” and “IQBt78Xrg4c60auc195lhkG8nJl2.”

When the  former of a Nurse UUID is clicked, the  changes to  and notification message IDs are listed as shown the upper part of Fig. 11 (b). When the  former of a notification message ID such as “L8WQVLzi3Vc989MUKfs” is clicked, the  changes to  and the contents of the notification are listed. The line “created_at” shows the coordinated universal time in Java that the notification has been sent, “instance_id” is the encrypted Patient UUID, “message” is a practical message for which the meaning is “Drip system for Murata has left,” and “title” is the patient’s name. The “message” and “title” lines are changed for each application.

When the  changes to  and former of a nurse UUID is clicked, the  changes to  and the patient UUIDs and tokens are listed as shown in the middle part of Fig. 11 (a). This tells us which sensor device each nurse has established.




When the  former of a patient UUID is clicked, the  changes to  and the lines “name”, “nurses” and “state” appear as shown in the lower part of Fig. 11 (a). The “name” line shows the patient’s name in this application. The line “nurses” shows the nurses who have read the QR-code with their mobile terminal. In this case, two nurses whose encrypted UUIDs are “4ANVMkv1kDarruVlt8l1opmAYj13” and “IQBt78Xrg4c60auc195lhkG-8nJl2” monitor two sets of sensor devices “csYUUM90CF0” and “dhwYz4dyGLw.”



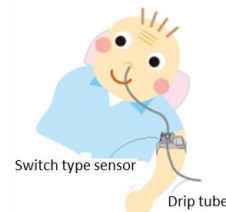
Figure 11. Examples of monitoring display.

D. Example application

We developed two monitoring applications that adopt the proposed notification system. One is an intravenous drip monitoring application; the other is a fall monitoring application. We describe their details in this section.

1) Intravenous drip monitoring

Some cognitive impairment patients sometimes remove an intravenous drip set by themselves. One existing intravenous drip monitoring system “Tenteki call” uses a switch type sensor that fastens a drip tube to detect when a drip set is removed as shown in Fig. 12 [16]. In case of a “Tenteki call”, if a patient removes the switch type sensor together with the drip set, the sensor cannot detect the removal.



(a) Switch type sensor. (b) Example of use.

Figure 12. “Tenteki” call (Japanese), Technos Japan Co., Ltd.



We use a magnetic patch and a wireless magnetic sensor to detect removing a drip set, as shown in Fig. 13. The magnet is fastened to the body with an adhesive film in a place such as an arm. An intravenous drip tube is also fastened to the wireless magnetic sensor and the sensor is fastened to a magnetic patch with a medical fixing film. When a patient removes an intravenous drip set, the sensor is also removed from the body part. However, since the magnetic patch is fastened to the body part with an adhesive film, it must remain on the body part.

We developed the prototype system shown in Fig. 14. We used the STEVAL-WESU1 [17] developed by STMicroelectronics as the wireless magnetic sensor (see Fig. 14 (a)), and an Android smartphone as the wireless relay unit and mobile terminal. The muscle stiffness obtained by equipment manufactured by PIP Co., Ltd. [18] was used as the magnetic patch (see Fig. 14(b)). In this prototype system, the program for detecting removal is integrated with the wireless relay program.

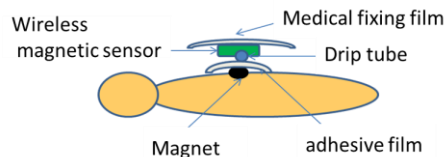


Figure 13. Structure of the magnetic intravenous drip monitoring



(a) Magnetic sensor and attached tube.



(b) Magnetic patch on an arm.



(c) Magnetic sensor on a magnetic patch.



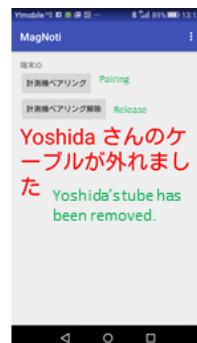
(d) A removed drip set.



(e) W. R. display : monitoring



(f) W. R. display : detecting a tube removed



(g) M. T. display : receiving a message

Figure 14. Drip monitoring system prototype.

While a magnetic sensor is on a magnetic patch (see Fig. 14 (c)), the measured magnetic strength is bigger than the decision level (see Fig. 14 (e)). When the magnetic sensor is removed (see Fig. 14 (d)), the measured magnetic strength is less than the decision level; the detecting program has determined that an intravenous drip set has been removed (see Fig. 14 (f)). The wireless relay program sends a message “Yoshida’s tube has been removed.” That message is displayed on the mobile terminal (see Fig. 14 (g)).

1) Fall monitoring

Elderly people, especially cognitive impairment patients, have an increased risk of falling and consequently injuring themselves. They need to be prevented from falling to maintain their health because injuries from falling are a major reason for them to prolong their staying in a hospital.

Therefore, many kinds of fall prevention systems have been developed. Most of them are classified into three schemes. The first type uses a mat type sensor like the systems described in Section II, the second one uses load sensors that are mounted in the legs of a bed, and the third one uses a camera. We developed a fall prevention system in which MS-KINECT was used. This is one of the third types. M. J. Rantz developed a fall detection system that uses MS-KINECT [19]. A medical worker monitors and judges whether a patient falls through the depth image of a patient on a monitor display.

On the other hand, our developed system detects whether a patient in a bed wakes up, sits up, stands up, or falls on the floor with a skeleton image of the patient. The detecting algorithms are as follows;

- (1) Waking up: detecting that the head’s height position is higher than the judging height 1.
- (2) Sitting up: detecting both shoulders and a spine base angle of 25 or more degrees.
- (3) Standing up: detecting the head, both shoulders and both hips, and a head is higher than the judging height 2.
- (4) Falling down: detecting that the head’s height position is lower than the judging height 3.

We experimentally tested whether the developed system can detect the four conditions given above. The MS KINECT was positioned diagonally in front of the bed so that the front of the patient could be observed as shown in Fig. 15. Experimental results are shown in Fig. 16. Since MS-KINECT works on a PC, the sensor device program that detects patient conditions with MS-KINECT and the wireless relay unit program are combined on a PC. Monitoring images of participants as patients and the QR-code for pairing are shown on the PC display.

The developed system can detect four conditions for a patient. With it, a mobile terminal receives a “standing up” message sent from a wireless relay unit. However, the system sometimes fails to detect conditions or does not detect them correctly. Accordingly, we plan to improve our

algorithms so that detection accuracy will be increased. We should also mention that a patient using this experimental system is presented with color images. We plan to change these color images to depth images to maintain a patient's privacy.

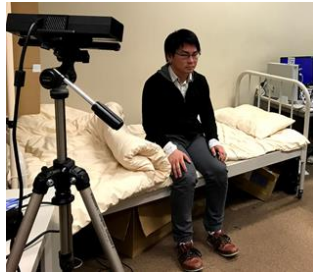
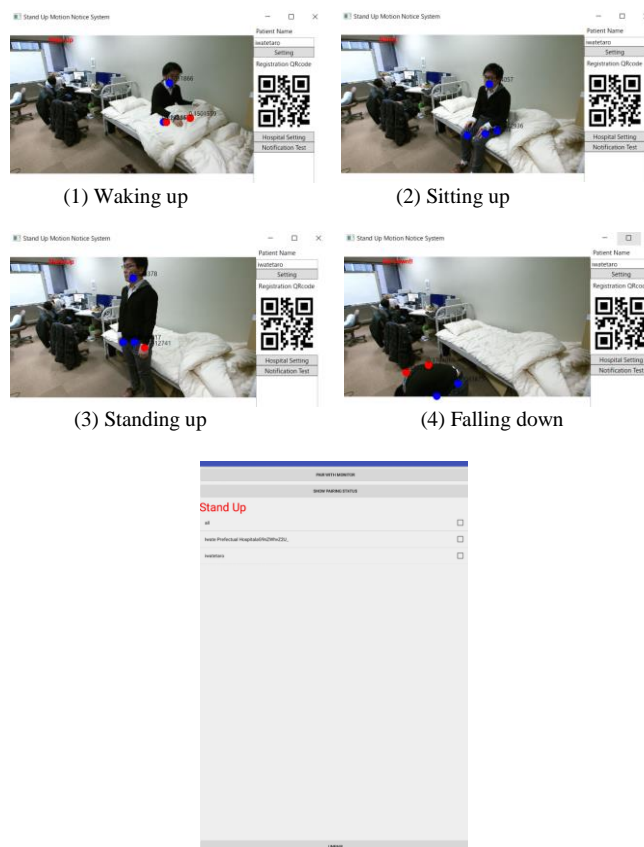


Figure 15. Experimental image.



(5) Screenshot of a mobile terminal that detects a “standing up” message

Figure 16. Experimental results for patient monitoring.

IV. DATA COLLECTING SYSTEM

In this section, we describe the data collecting system for data measuring applications.

A. Design concept

We designed the system so that:

- (1) A medical person could easily install a measuring device at a bedside.
- (2) Measured data were stored for each patient and the corresponding medical professional in the cloud server.
- (3) A medical professional can access only authorized data.
- (4) A supervisor can easily change the relationship among patients, nurses who install measuring devices and medical professionals who analyze data through remote terminals.

Therefore, no operations are performed with the console terminal and measuring devices can be easily installed at the patient's bedside. The other side, the medical professional such as a medical doctor in charge, can download measured data files to keep security and safety.

The system configuration is shown in Fig. 17. Before starting to use the system, the supervisor inputs login data that comprises the UID and password for a medical professional. Measured data are stored as files in the sensor relay unit at first. The files are classified for each patient on the accessed web page. When medical professionals access the given URL and login data with their recorded UID and password, a specified page for each medical professional is presented. The medical professional clicks the file and analyzes it.

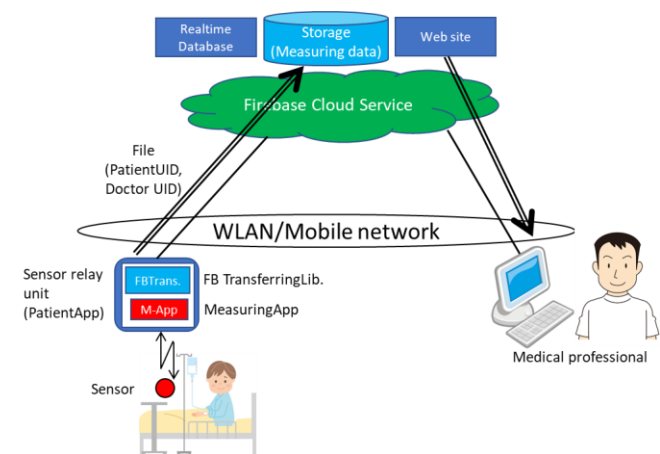


Figure 17. Configuration of the data collecting system.

B. System configuration

The data collecting system, also designed on Firebase, contains the following functions:

- (1) Measured data upload function: The sensor relay unit measures medical data using sensors and uploads the data to Firebase as files.
- (2) Measured data download function: A medical professional accesses Firebase and downloads medical data files.

We developed a PatientApp program that works on the sensor relay unit and a WebSite program that works on Firebase to provide these functions. The PatientApp uploads the measured data file to the Storage on Firebase. The WebSite collaborates with the Storage and provides a file download function to a medical professional through the Web browser. In this subsection, we introduce sequence flows that provide these functions.

1) Measured data upload function (Fig. 18)

PatientApp consists of “MeasuringApp” and “FB TransferringLib” programs. The MeasuringApp program provides the User interface needed to login, enter information, measure medical data and create a file. The FB Transferring Lib. Program uploads the file to the Storage in GFB.

Before developing a PatientApp program, a developer accesses Firebase to download a configuration file. An Android application package file (Apk File) is then made as a building application and connects to Firebase as shown in Fig. 4. These steps are the same as those in the building application given in subsection III_B_1). Before using an Android terminal as the sensor relay unit, the PatientApp in the sensor relay unit must create its own account in the Realtime Database with the Anonymous certification in the same project. After a measured file has been made, the PatientApp uploads the file to the storage server in Firebase as shown in Fig. 18. The storage server generates the file download URL, which is managed in the Realtime Database.

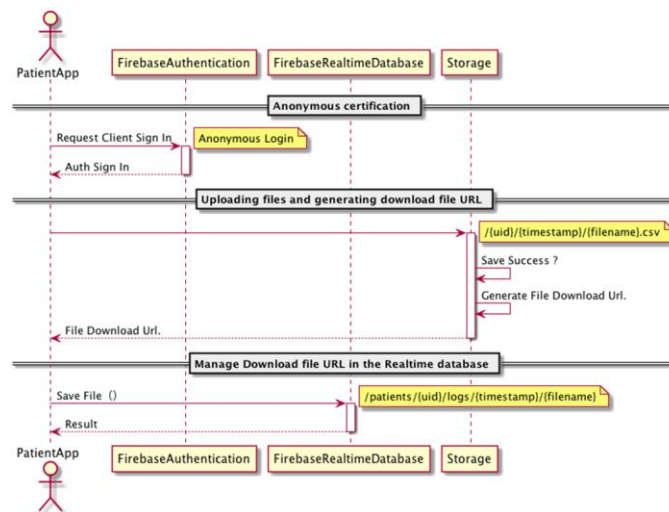


Figure 18. Sequence flow to upload measured files.

2) Measured data download function (Fig. 19)

Supervisors input the access account of medical professionals from the management page in Firebase. The sequence flow with which medical professionals download their patients' files is shown in Fig. 19. When medical professionals access the Website, they log in with their assigned ID and password.

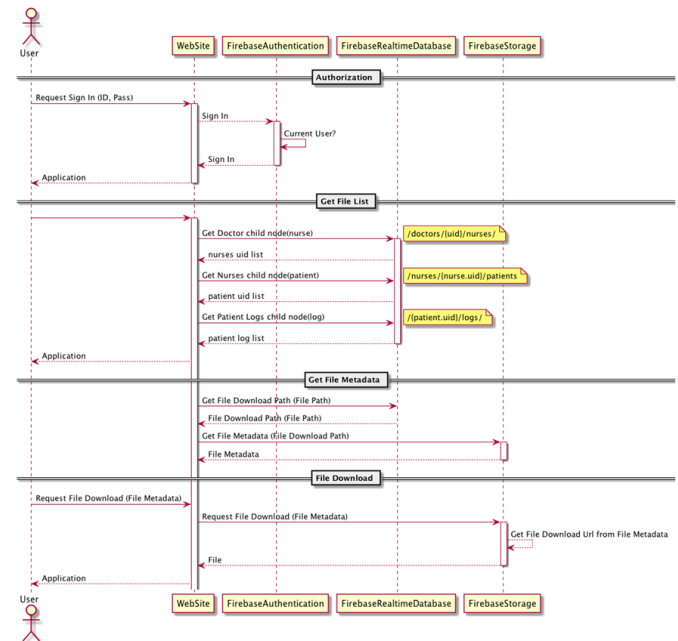


Figure 19. Sequence flow to download measured files

After login, the Website application accesses the Realtime Database to get information related to nurses and patients and presents them. The Website application also gets meta-data such as an access path to a stored file. When a medical professional clicks a file on the web page, the Website application accesses the indicated file on the Storage through the access path. Finally, the indicated file is downloaded.

C. Example application

We developed a measuring application that uses the above data collecting system. In this application, motions of the wrist and lumbar region were measured with a 3D-acceleration sensor. We used two SONY Smart Watch 3 units as the 3-D acceleration sensor [20]. One was attached to the wrist with a wrist band and the other was attached to the lumbar region with a lumbago band as shown in Fig. 20. Both were connected to a sensor relay unit with Bluetooth. A patient's name is input and then the body parts corresponding to sensors are input. In this case, we input “Wrist” and “Lumbar.” After inputting above words, the page shown in Fig. 21 has been presented. There are four buttons in this page. “QR CODE” button is used to make a connection with a nurse. Clicking “START” button starts measuring and clicking “STOP” button stops measuring. Clicking “TRANSFER” uploads a measured data file to the storage server in Firebase.

After a medical professional logged onto the WebSite program, the patients' names and measured data files appeared as shown in Fig. 22. Clicking the name of a file downloads the file.

Sample graphs of measured data for the wrist and lumbar region are shown in Fig 23. The orange (Wrist X),

gray (Wrist Y) and blue (Wrist Z) lines show respectively the data obtained when the arm moved up/down, forward/backward and in a twisting motion. The yellow (Lumbar X), blue (Lumbar Y) and green (Lumbar Z) dot-lines show respectively the antelexion, lean and twist of the upper body. In this example, the participants used chopsticks to eat. When eating, they first twisted their arm so that it faced the mouth. Then they moved the arm up to the mouth while simultaneously leaning the upper body forward a little. After putting food into the mouth, they moved the arm back to where it had been. However, they continued to lean forward while eating.

We will release further details of this application in the near future.



Figure 20. Participant having two sets of SONY Smart Watch 3

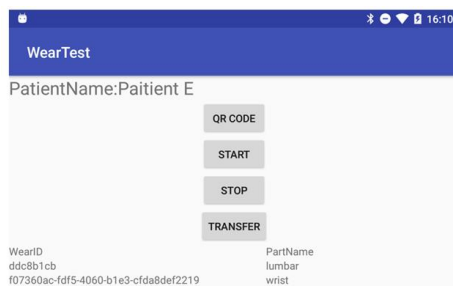


Figure 21. User interface of PatientApp

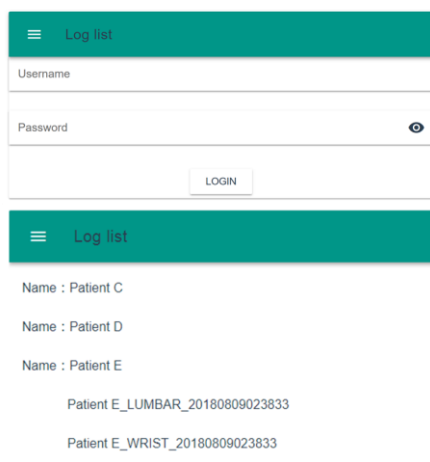


Figure 22. WebSite user interface.

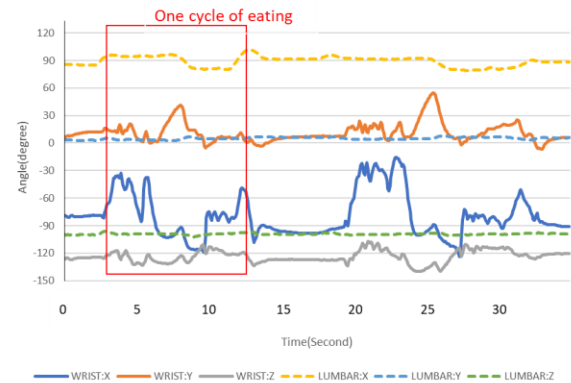


Figure 23. Sample graphs of measured data for the wrist and lumbar region.

V. CONCLUSION

We have developed and propose a novel patient monitoring platform for healthcare facilities. The easy to use platform consists of a notification system for event monitoring applications and a data collecting system for data measuring applications. In the notification system, monitoring devices are paired with mobile terminals by reading a Quick Response Code on monitoring terminal (sensor relay unit) displays. It is possible to pair them at the bedside of a patient without a console terminal and operator. When the sensor devices detect a patient having trouble, event messages are sent to a mobile terminal, not to a nurse station. Therefore, no operations are performed with the console terminal and medical devices can be easily installed at the patient's bedside.

We also developed a data collecting system with which medical data files measured by sensor devices are automatically uploaded to a cloud server and can be downloaded from the server by authorized medical professionals. No operations are performed with the console terminal in this system.

It is possible to develop monitoring or measuring application using the proposed platform and several manufactured sensors. This means that our platform keeps interoperability.

We have proposed to jointly develop commercial version of the proposed platform and/or applications to several companies. Some companies are interested, however, we do not have yet contracts with these companies.

ACKNOWLEDGEMENT

Thanks to Kazuhiro Yoshida for help in performing this research. This research and development work was supported by the MIC/SCOPE #181602007.

REFERENCES

- [1] Y. Murata, R. Takahashi, T. Yamato, S. Yoshida, and M. Okamura, "Proposal of Field Oriented Event Messaging System," IARIA, eTELEMED 2018, pp. 16-20, March 2018.
- [2] Nurse Call BCAC, <https://www.carecom.jp/global/solutions/nurse-call-bcac/> [retrieved: November, 2018].
- [3] Mat sensor, Carecom, <https://www.carecom.jp/global/solutions/option/> [retrieved: November, 2018].
- [4] Clino Guard, Ackermann, Honeywell, <https://www.ackermann-clino.com/en/products/tracking-localization/> [retrieved: November, 2018].
- [5] Patient Monitors, GE, http://www3.gehealthcare.com/en/products/categories/patient_monitoring/patient_monitors [retrieved: November, 2018].
- [6] QR-code, <http://www.qrcode.com/en/>, [retrieved: November, 2018].
- [7] Risho catch, Paramount Bed, <http://www.paramount.co.jp/learn/inversion/leaving> [in Japanese, retrieved: November, 2018].
- [8] H. U. Balaguera et al., "Using a Medical Intranet of Things System to Prevent Bed Falls in an Acute Care Hospital: A Pilot Study," Journal of Medical Internet Research, <https://www.jmir.org/2017/5/e150/>, [retrieved: November, 2018].
- [9] A. V. Halteren et al., "Mobile Patient Monitoring: The MobiHealth System," The Journal on Information Technology in Healthcare 2004, vol. 2(5), pp.365–373, 2004.
- [10] Y. Lin et al., "A Wireless PDA-Based Physiological Monitoring System for Patient Transport," IEEE Transactions on Information Technology in Biomedicine, Vol. 8, No. 4, pp.439-447, 2004.
- [11] T. Gao et al., "Vital Signs Monitoring and Patient Tracking Over a Wireless Network," Johns Hopkins APL Technical Digest, Vol. 27, No. 1, pp. 66-74, 2006.
- [12] P. Várady, Z. Benyó, and B. Benyó, "An Open Architecture Patient Monitoring System Using Standard Technologies," IEEE Transactions on Information Technology in Biomedicine, Vol. 6, No. 1, pp. 95-98, 2002.
- [13] Patient monitoring, GE Healthcare, http://www3.gehealthcare.com/en/products/categories/patient_monitoring/, [retrieved: November, 2018].
- [14] Google Firebase Cloud Messaging, <https://firebase.google.com/docs/cloud-messaging/?hl=en>, [retrieved: November, 2018].
- [15] Open Source QR Code Library, <http://qrcode.osdn.jp/index.html.en>, [retrieved: November, 2018].
- [16] Tenteki call (Japanese), Technos Japan Co., Ltd., <http://www.technosjapan.jp/product/infusion/2012/0121195451.html> [in Japanese, retrieved: November, 2018].
- [17] STEVAL-WESU1, STMicroelectronics, <http://www.st.com/en/evaluation-tools/steval-wesu1.html> [retrieved: November, 2018].
- [18] Muscle Stiffness, <http://www.pip-club.com/english/muscle.html>, [retrieved: November, 2018].
- [19] M. J. Rantz et al., "Automated Fall Detection With Quality Improvement "Rewind" to Reduce Falls in Hospital Rooms," NIH Public Access, Journal of Gerontological Nursing, 40(1), pp. 13–17, 2014.
- [20] SONY Smart Watch 3, <https://www.sonymobile.com/global-en/products/smart-products/smartwatch-3-swr50/#gref>, [retrieved: November, 2018].

Compositing “Stand Off” Ground Penetrating Radar Scans of Differing Frequencies

Roger Tilley, Hamid R. Sadjadpour, Farid Dowla

Department of Electrical Engineering

University of California, Santa Cruz

Santa Cruz, CA. 95064

Email: {rtvax, hamid, dowla} @soe.ucsc.edu

Abstract— Methods have been developed to combine signals of various frequencies in a manner to produce clearer images in the presence of noise. Ground Penetrating Radar (GPR) scans at various frequencies are no exception. Methods using an optimization problem solver, the Expectation-Maximization (EM) Algorithm, define weights used to perform the task of combining GPR scans. In this paper, we explore using the Gaussian Mixture Model (GMM) feature of the EM Algorithm on GPR scans taken at various heights above ground (“Stand Off” GPR). This method demonstrates the same measured improvement toward producing a cleaner image as GPR scans taken at ground level using the same EM Algorithm method.

Keywords—Ground Penetrating Radar; Expectation Maximization; Gaussian Mixture Model; Maximum Likelihood parameter estimation; Finite Difference Time Domain Method, GprMax.

I. INTRODUCTION

Illuminating objects at various depths in a variety of terrains is the purview of Ground Penetrating Radar (GPR) scans. Different frequencies illuminate best at different depths. The higher the frequency the better illuminated the objects close to the surface are, with great fidelity. Conversely, the lower the frequency the better objects are illuminated at lower depths but with less detail. We previously examined, treating GPR scans at several frequencies, over the same area, like sub-components of a square wave. Where summing these sub-components, weighted by magnitude, created a square wave; suggesting that summing GPR Scans should form a crisper image to a lower depth than any one frequency scan. We reported that simply adding each scan together, as shown in [1][2], does not suffice. Summing weighted versions of each scan presents the best solution [1]. Determining the weights of each frequency scan provided the challenge. We were able to show that the EM Algorithm, an optimization problem solver, was an effective method for combining ground based GPR scans, using the data mixture feature [1]. What remains to be explored is whether the same is true for GPR scans at varying heights above the ground for the same buried objects and media, we previously examined [1].

In this paper, we explore the use of the Expectation Maximization (EM) Algorithm [3] in a role as an

optimization problem solver to determine the weights to be applied to each scan for an optimal weighted combination of scans. In Section II we discuss related work to combining GPR scans and processing them at varying heights. In Section III, we describe the EM data mixture process and its data mixture feature as it pertains to GPR scanned data.. In Section IV, we briefly cover the Maximum-Likelihood (ML) Estimation process as related to the EM Algorithm data mixture process [3][4]. In Section V, we present results of compositing of simulated GPR scan examples using the GprMax [5] software program to develop the individual GPR scans at various frequencies and transmitter/receiver heights above ground with buried targets in a defined media. In Section VI we draw conclusions and discuss future work.

II. RELATED WORK

A literature search on compositing of GPR signals revealed only a few works. Papers on GPR time-slice analysis, overlay analysis and GPR isosurface rendering mostly by archaeologists; all similar in approach, were found. The technique was to illuminate the strongest reflections with a color or shading then combine the information by layers of depth, displaying the result [6].

For compositing of ground based GPR scans, work by Dougherty et al. [7] was the earliest found. Dougherty focused on methods to align each trace at its direct arrival peak, subtract the direct arrival pulse, and equalize the magnitude weight of each trace, then combine the traces. Booth et al. [8][9] confirmed Dougherty et al. [7] results and presented improved methods to develop frequency scan weighting techniques based on trace averaged amplitude spectra. Booth et al. [8][9] also proposed time invariant weighting methods consisting of matching the compositing results to an idealized amplitude spectrum output from a least-squares analysis.

Bancroft [10] introduced a double ramp summation method for computing weights. One ramp suppresses a frequency's energy while another ramp increases an adjacent frequency's energy all over time. The length of the ramp and start time was based on the wavelength of the frequency of interest. Weights developed as a ratio of the

average envelope of GPR frequencies was Bancroft's [10] additional contribution. Improvements over the previous works were minimal. This result is discussed in more detail in references [1][2].

A brief look at compositing of GPR scans at various heights revealed even less information. The literature at various heights was mostly concerned with Synthetic Aperture Radar (SAR) using single frequencies or chirped frequencies. The SAR literature focused on compensating for geometric distortion as the radar traversed the scan area; developing methods to piece the individual scans together. Other SAR papers [11][12][13] concerned themselves with accounting for phase shifts in the data from the angles the radar signals were sent and received from. Much discussion revolved around Gazdag [14], Stolt and FK migration [15] techniques. These techniques are beyond the scope of this paper and are part of future work discussions. Another SAR method discussed using Ultra-Wideband SAR radars to distinguish buried objects using an author developed "Method of Moments algorithm" [11].

Most, directly, related work focused on mathematically defining weights for each frequency by equal weighting, or defining weights that equalized the spectra of GPR frequencies through ramp summation, or a least squares process matching an idealized amplitude spectrum. Our previous work [1][2] explored GPR scans as a cluster mixture model problem using EM optimization problem solving methods for ground based scanned objects.

III. EXPECTATION MAXIMIZATION ALGORITHM

To group like items contained in complex mixtures, or to solve incomplete data problems, or to determine membership weights of a collection of data points in a cluster, all are types of problems considered the purview of the EM Algorithm solution process. Our compositing of GPR scans process exploits this last feature, determining the membership weights in a cluster of data points within a Gaussian Mixture Model (GMM) [16][17]. The Gaussian distribution was chosen over other distributions because it is often used when the distribution of real-valued random variables is unknown.

We can define the EM Algorithm GMM process by first defining a finite mixture model, $f(\underline{x}; \theta)$, of K components as mixtures of the following GMM function:

$$f(\underline{x}; \theta) = \sum_{k=1}^K \alpha_k p_k(\underline{x} | \theta_k), \quad (1)$$

Where:

- $p_k(\underline{x} | \theta_k)$ are K mixture components with a distribution defined over $p(\underline{x} | \theta_k)$ with parameters $\theta_k = \{\underline{\mu}_k, C_k\}$ (mean, covariance)
- $p_k(\underline{x} | \theta_k) =$

$$\frac{1}{(2\pi)^{d/2} |C_k|^{1/2}} e^{-\frac{1}{2}(\underline{x} - \underline{\mu}_k)^T C_k^{-1} (\underline{x} - \underline{\mu}_k)} \quad (2)$$

- α_k are K mixture weights, where $\sum_{k=1}^K \alpha_k = 1$.
- $\{\underline{x}_1, \dots, \dots, \underline{x}_N\}$ Data set for a mixture component in d dimensional space.

The EM Algorithm has 2 steps for each iteration. The first step is the Expectation step (E-step). The E-step determines the conditional expectation of the group membership weights (w_{ik} 's) for \underline{x}_i 's, introducing unobservable data based on θ_k , the mean and covariance matrix. The second step is the Maximization step (M-step). New parameter values ($\alpha_k, \underline{\mu}_k, C_k$); mixture weights, mean and covariance of weights; to maximize the finite mixture model are computed. The E-step and M-Step are repeated until convergence of the GMM model is reached. Convergence is characterized as the minimal change of the log-likelihood of the GMM function from one iteration to the next. The E-step and M-step equations are defined below:

E-Step –

$$w_{ik} = \frac{p_k(\underline{x}_i | \theta_k) \alpha_k}{\sum_{m=1}^K p_m(\underline{x}_i | \theta_m) \alpha_m} \quad (3)$$

for $1 \leq k \leq K, 1 \leq i \leq N$;

with constraint $\sum_{k=1}^K w_{ik} = 1$

M-Step –

$$N_k = \sum_{i=1}^N w_{ik} \quad (4)$$

$$\alpha_k^{new} = \frac{N_k}{N}, \text{ for } 1 \leq k \leq K \quad (5)$$

$$\underline{\mu}_k^{new} = \left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} * \underline{x}_i \quad (6)$$

for $1 \leq k \leq K$

$$C_k^{new} =$$

$$\left(\frac{1}{N_k} \right) \sum_{i=1}^N w_{ik} * (\underline{x}_i - \underline{\mu}_k^{new}) (\underline{x}_i - \underline{\mu}_k^{new})^T \quad (7)$$

Convergence (log likelihood of $f(\underline{x}; \theta)$) –

$$\text{Log } l(\vartheta) =$$

$$\sum_{i=1}^N \log f(\underline{x}_i; \theta) =$$

$$\sum_{i=1}^N (\log \sum_{k=1}^K \alpha_k p_k(\underline{x}_i | \theta_k)) \quad (8)$$

These equations were implemented in MATLAB. The different scanning frequencies are represented by the variable 'k' and 'x' represents the GPR trace scans. Each trace, at a frequency and transmitter (Tx)/receiver (Rx)

position, are analyzed and combined for all frequencies before moving on to the next position. The EM GMM processing steps are briefly outlined below:

Expectation Maximization Gaussian Mixture Model process:

1. Initialize algorithm parameters; weights (mixture and group membership), mean, covariance, for each trace.
2. Expectation step – estimate parameters.
3. Maximization step – maximize estimated parameters.
4. Check for convergence – log likelihood of mixture model.
5. Repeat steps 2 – 4 until change from iteration to iteration is below or equal a defined value.
6. Combine traces with defined mixture weights.

IV. MAXIMUM LIKELIHOOD ESTIMATION PROCESS AND THE EM RELATIONSHIP

The EM algorithm provides a way to reduce a Maximum Likelihood Estimation (MLE) problem to a simpler optimization sub-problem, which is guaranteed to converge. This is the relationship between the MLE process and the EM algorithm. The MLE process provides an estimate of the unknown parameter, which maximizes the probability of getting the data we observed (likelihood).

An MLE process can be described as follows. Given a random sample X_1, X_2, \dots, X_n , independent and identically distributed (i.i.d.) with a probability density function $f(x_i, \theta)$, where θ is the unknown parameter to be estimated; the joint probability density function (PDF) can be labeled as $L(\theta)$.

$$L(\theta) = P(X_1 = x_1, X_2 = x_2, \dots, X_n = x_n) = f(x_1; \theta) * f(x_2; \theta) \dots f(x_n; \theta) = \prod_{i=1}^n f(x_i; \theta) \quad (9)$$

Should the probability density function (PDF) be Gaussian with known variance σ^2 and unknown mean, μ , then, the likelihood equation becomes the following:

$$L(\mu) = \prod_{i=1}^n f(x_i; \mu, \sigma^2) = \sigma^{-n} (2\pi)^{-n/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^n (x_i - \mu)^2\right) \quad (10)$$

To determine the maximum value of the parameter μ , for the likelihood equation, take the partial derivative of the log-likelihood equation with respect to (w.r.t.) the mean, μ , and set the result equal to 0 and solve the remaining equation for the variable μ . To determine if this value represents a maximum value for the likelihood equation, take the second partial derivative of the log-likelihood equation w.r.t. μ ; should a negative value result, this verifies the parameter μ , found is a maximum value for the likelihood function.

$$\text{Log}(L(\mu)) = -n \log(\sigma) - \frac{n}{2} \log(2\pi) - \sum_{i=1}^n \frac{(x_i - \mu)^2}{2\sigma^2} \quad (11)$$

$$\frac{\partial}{\partial \mu} (\log(L(u))) = -2(-1) \sum_{i=1}^n \frac{(x_i - \mu)}{2\sigma^2} = 0 \quad (12)$$

$$\text{Solve for } \mu; \quad \mu = \frac{\sum_{i=1}^n x_i}{n} \quad (13)$$

Second derivative –

$$\frac{\partial^2}{\partial \mu} (\log(L(u))) = \sum_{i=1}^n \frac{(-1)}{\sigma^2} = -\frac{n}{\sigma^2} \quad (14)$$

The second derivative is negative; by definition the calculated μ value is a maximum.

When there are at least two sets of data, one partially observed (hidden), or when mixture parameters are to be estimated, the MLE process becomes hard. For example, a mixture distribution of the form $f(x) = \sum_{k=1}^K \alpha_k f(x; \theta_k)$, where there are K number of components in the mixture model and for each k, there is a PDF, $f(x; \theta_k)$ with weights α_k and a complete observed data set x with additional constraints $\sum_k \alpha_k = 1$ and $\alpha_k \geq 0$ for all k; the joint PDF has the following form with n observed data for each k:

$$L(x|\alpha, \theta_k) = \prod_{i=1}^n \sum_{k=1}^K \alpha_k f(x_i; \theta_k) \quad (15)$$

The log of the likelihood equation is as follows:

$$\text{Log}(L(x|\alpha, \theta_k)) = \sum_{i=1}^n \log \sum_{k=1}^K \alpha_k f(x_i; \theta_k) \quad (16)$$

Using MLE to solve this equation presents a challenge to determine the derivative of the log of sums and the start value for the weight, α_k , associated with an individual distribution. Many local maxima can be found that are less than the global maximum, calculated using an established value of α_k . Selecting the weight value that attains the global maximum for the above log-likelihood equation is not likely in short order.

The EM algorithm process provides a method to estimate the weights, guarantee convergence of the log-likelihood equation [3][4] to a non-decreasing local maximum with each completion of all steps outlined in Section II. A feature of the EM algorithm is that each local maximum achieved increases toward a global maximum. The E-step uses existing values to calculate the probability of weight start values. The M-step recalculates the model parameters then, calculates the maxima for that set of parameters using the MLE process. The EM algorithm reduces the MLE optimization problem to a sequence of simpler optimization sub-problems, each guaranteed to converge. The EM process is repeated until a global maximum is reached.

The EM algorithm incorporates the MLE process only after reducing the model to a form, which is guaranteed to converge. To combine GPR frequency scans, the actual weights of each frequency scan are unknown or hidden. The manner the EM algorithm uses to accomplish workable solutions to hidden or incomplete data, makes a distinction from other optimization problem solvers; as a result, making

it a featured candidate to provide a solution to combining multiple GPR frequency scans.

V. GPR SCAN RESULTS

We examine extending the capability of the EM GMM problem solver by using the defined model areas from our previous work [1][2]. Areas were defined using the Finite Difference Time Domain (FDTD) [18] modeling software package GprMax by A. Giannopoulos [5] to create GPR scans in various media. 3-D hardware verification of the software was determined in reference [19], but only 2-D analyses were performed here. Examples were constructed such that the Transmitter (Tx) and Receiver (Rx) heights above the ground were changed for each EM GMM in-depth analysis. Tx/Rx heights examined included 5 meters, 10 meters, 20 meters and 40 meters.

The first defined area modeled consisted of Tx/Rx suspended 5 meters above the ground in air [1], repeated here for continuity in our discussion of height effects on EM GMM problem solver. The target (a perfect electrical conductor) is buried 10 meters below the surface in a moist-sand medium with relative permittivity (ϵ_r) of 9.0, and an electrical conductivity of 0.001 mS/m (milli-Siemens per meter) (Simulated Analysis 1 – SA1). The target is 2 meters length and 0.5 meters in depth. Each Tx/Rx is moved along the scan axis (x – axis) 0.25 meters per step for a total of 36 scans. The model area is 10 meters in width and 25 meters in depth. The Tx position starts at 0.5 meters ending at 9.5 meters, and the Rx position starts at 0.75 meters ending at 9.75 meters. Each scan is 425 ns in length, long enough to receive a reflected signal 24 meters below a Tx/Rx in the medium of air and moist-sand. The defined model has a minimum grid space of 200 points in the x direction, ($\Delta x = 0.05$ meters), and 500 points in the y-direction, ($\Delta y = 0.05$ meters). Figure 1 shows the model, Tx/Rx positions and target area.

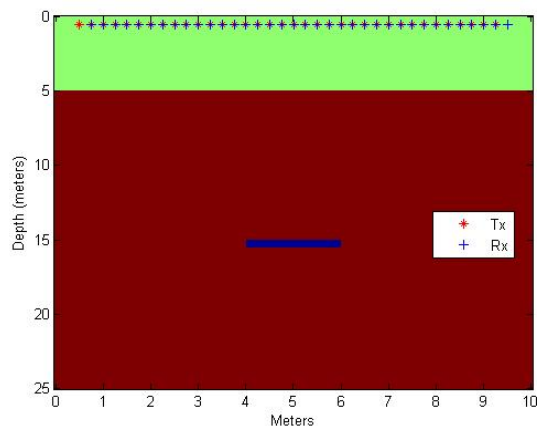


Figure 1. Defined Space with buried target at 15 meters depth and Tx's & Rx's 5 meters above ground.

Six frequency scans were calculated for model SA1. Scans at 20, 30, 50, 100, 500 and 900 MHz were combined using the EM GMM problem solver to determine the weights of each scan. Figure 2 shows the signals combined by scaling each signal max value to the same magnitude with the direct arrival and ground bounce signals removed. The target reflection is a broad area roughly 240 ns to 320 ns in depth (two-way travel time); a coarse indication of the depth of the target. The direct arrival signal is a signal that travels directly (line of sight) from a Tx to an Rx. A ground bounce signal is a radar return from the ground. The direct arrival signal was removed by subtracting a GPR scan without a target from a scan with a target, for each frequency. A broad area of target reflection is shown from approximately 240 ns to 320 ns in depth (two-way travel time); a very rough indication of target depth.

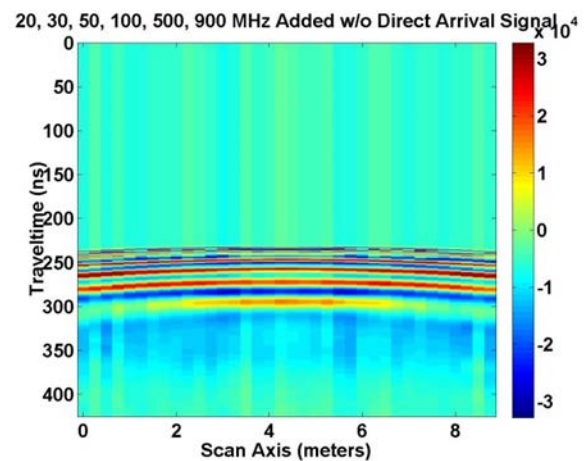


Figure 2. Sum of frequency signals with direct arrival and ground bounce signals removed.

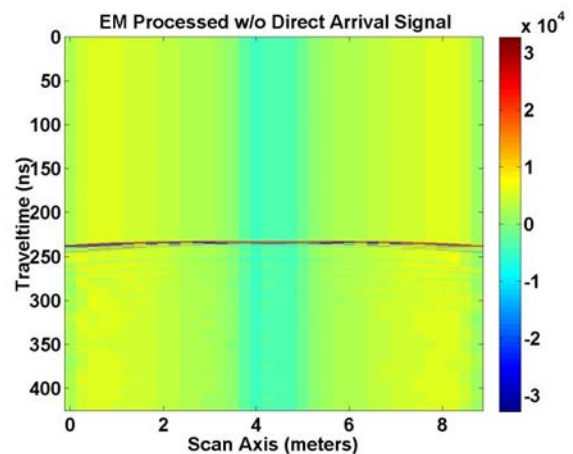


Figure 3. EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

Figure 3 shows the result of EM processed combined signals with the direct arrival and ground bounce signals removed. The target is correctly depicted at 10 meters below ground, approximately 15 meters below Tx's and Rx's (240 ns). The improvement of defining scan weights using the

EM Algorithm process is clearly visible. However, the Figure shows a broadened output result; broader than the target. This is attributed to the fact that the model is more like a bore hole; area twice as deep as its width. This and the inclusion of lower frequencies in the sum, account for the reverse “u-shaped” area beginning at the target depth outward.

Though the model created a bore hole effect, analysis continued for heights of 10, 20 and 40 meters above the ground with the same width model to judge whether the target would be revealed at the correct depth ignoring the target width that might be displayed. Figures 4–9 depict the model and the output result of combining 6 frequencies (20, 30, 50, 100, 500 and 900 MHz), using the EM algorithm to define the weights of each scan for each of the remaining 3 heights.

Figure 4 and Figure 5 depict the simulated analysis model and ground penetrating radar response when the Tx/Rx height is 10 meters above the ground. The target is correctly depicted at 20 meters (270 ns – two-way travel time) below Tx’s and Rx’s.

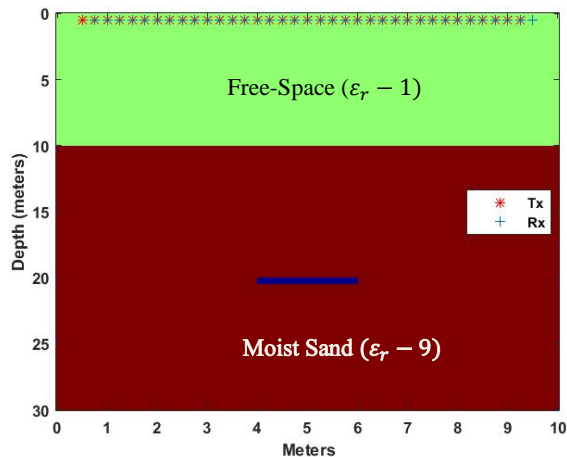


Figure 4. Defined Space of SA1, with buried target at 20 meters depth from Tx's & Rx's 10 meters above ground.

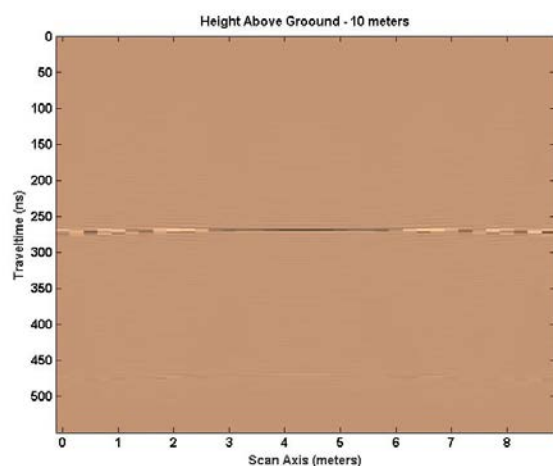


Figure 5. Output response of SA1, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

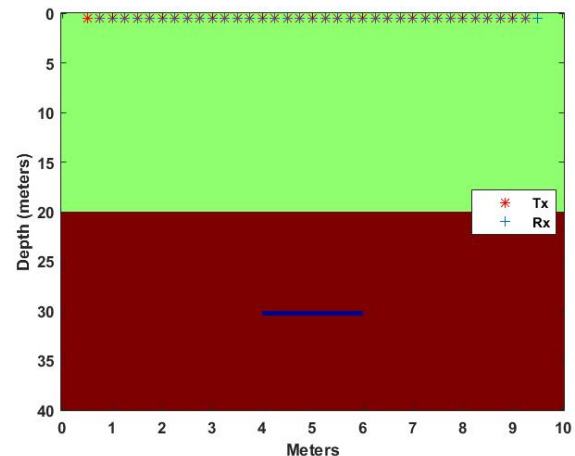


Figure 6. Defined space of SA1, with buried target at 30 meters from Tx's & Rx's 20 meters above ground.

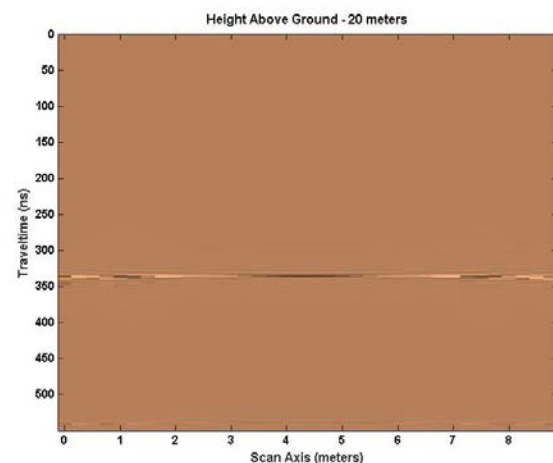


Figure 7. Output response of SA1, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

The simulated analysis of model SA1 was repeated for Tx/Rx heights of 20 meters above the ground (Figure 6) and 40 meters above ground (Figure 8). The target is depicted at 30 meters (335 ns) in Figure 7 and 50 meters (468 ns) in Figure 9 from the Tx's and Rx's, as expected for two-way travel times.

A second defined space model; (Simulated Analysis 2 – SA2) was developed to examine EM Algorithm response to a slightly different model type (Figure 10.). SA2 consists of an area 30 meters in length and 25 meters in depth. Four cases of Tx/Rx heights above the ground were analyzed. Cases included Tx/Rx at 5, 10, 20, and 40 meters above the ground. As before, a Tx/Rx combination is swept along the x direction axis beginning at 0.5 meters ending at 29.85 meters with Tx/Rx spacing of 0.25 meters. The number of GPR scans is 145 with a minimum grid space of 150 points in the x direction, ($\Delta x = 0.2$ meters) and 2500 points in the y direction, ($\Delta y = 0.01$ meters), dependent on the height above ground. The space above ground was defined as free-space with relative permittivity (ϵ_r) of 1.0 and electrical

conductivity of 0 mS/m or lossless. The scanned medium is dry-sand with a relative permittivity (ϵ_r) of 3.0 and electrical conductivity of 0.01 mS/m.

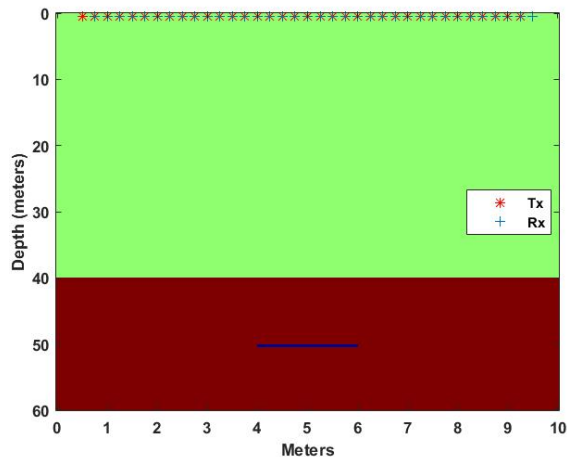


Figure 8. Defined Space of SA1, with buried target at 50 meters depth from Tx's & Rx's 40 meters above ground.

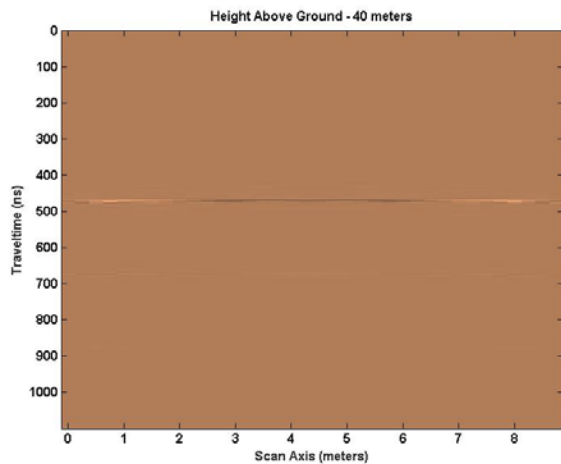


Figure 9. Output Response of SA1, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

Buried in the ground at 8 different levels (4.565, 6.065, 8.565, 10.065, 12.815, 14.065, 16.565 and 18.065 meters) are sheets modelled as perfect electrical conductors [1][2]. Each sheet is 2 meters in length and 0.1 meter thick. The scanning frequencies are the same as previously noted, (20, 30, 50, 100, 500 and 900 MHz). Figure 10, Figure 12, Figure 14 and Figure 16 show the four SA2 models for Tx/Rx heights above ground (5, 10, 20 and 40 meters) and the simulated sheets of corrugated aluminum modelled as perfect electrical conductors. References [1][2], used the same model with the exception that the Tx's and Rx's were just barely above ground. Figure 11, Figure 13, Figure 15, Figure 17 and Figure 18 display the GPR response after being processed using the EM GMM algorithm. Figure 18 displays the individual GPR traces instead of the image response. The direct arrival and ground bounce signal have been removed by subtraction in each case. At each height 8 sheets are depicted though their outline is not very clear and

worsens as the height increases. At 40 meters, only the individual trace response designates the 8 sheets.

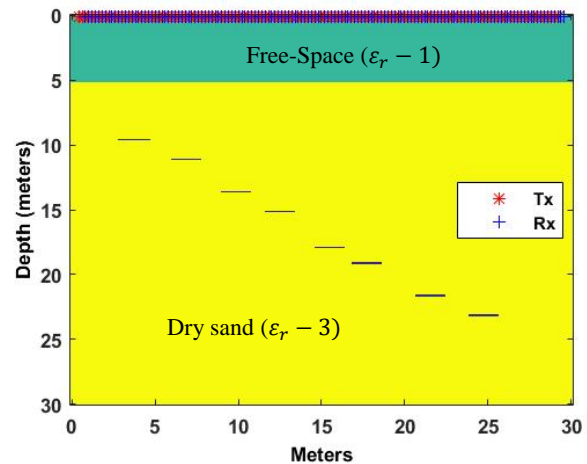


Figure 10. Defined Space of SA2, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 5 meters above ground.

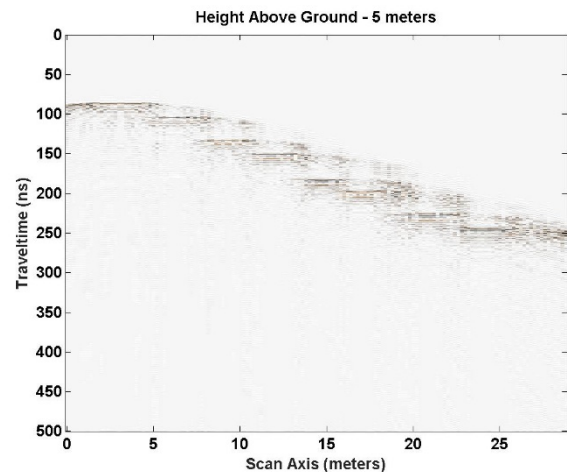


Figure 11. Output response of SA2, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

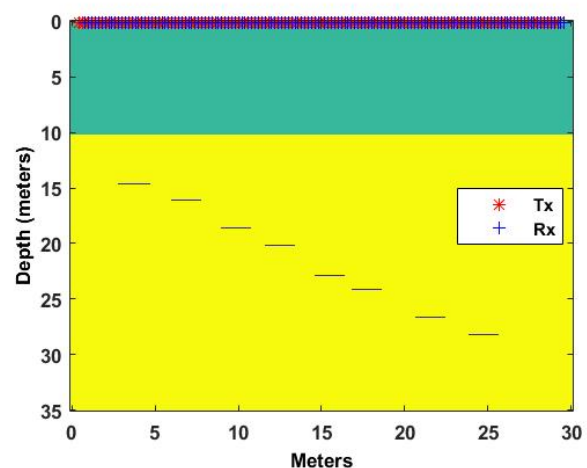


Figure 12. Defined Space of SA2, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 10 meters above ground.

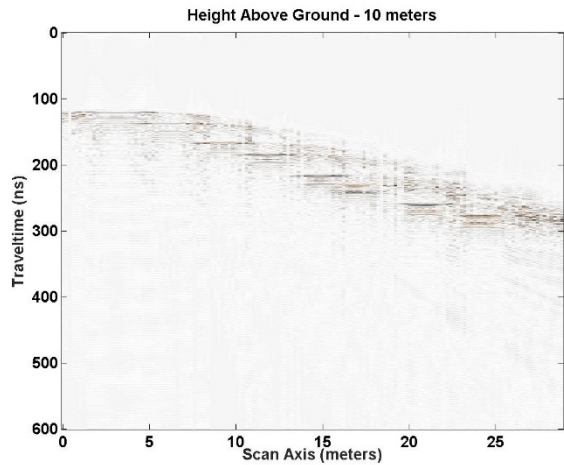


Figure 13. Output response of SA2, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

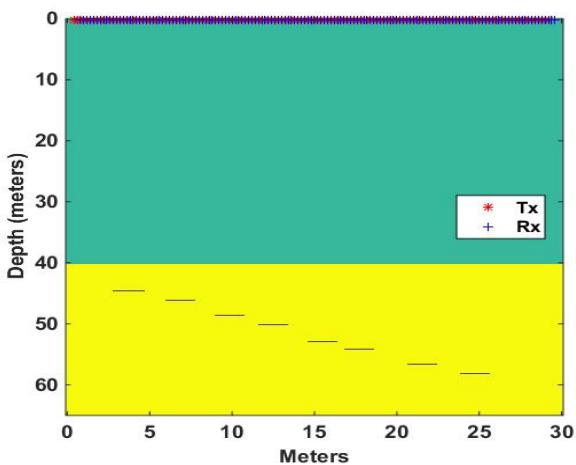


Figure 16. Defined Space of SA2, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 40 meters above ground.

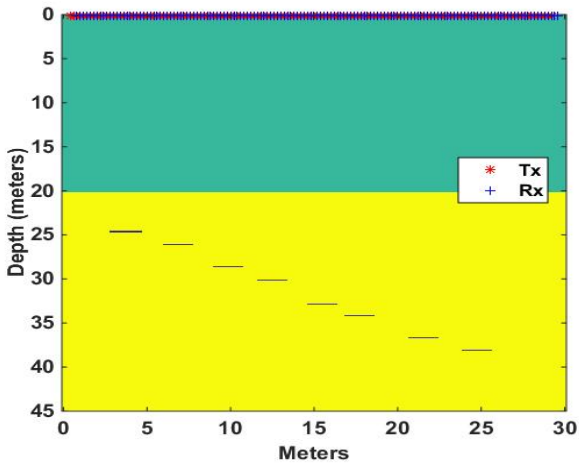


Figure 14. Defined Space of SA2, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 20 meters above ground.

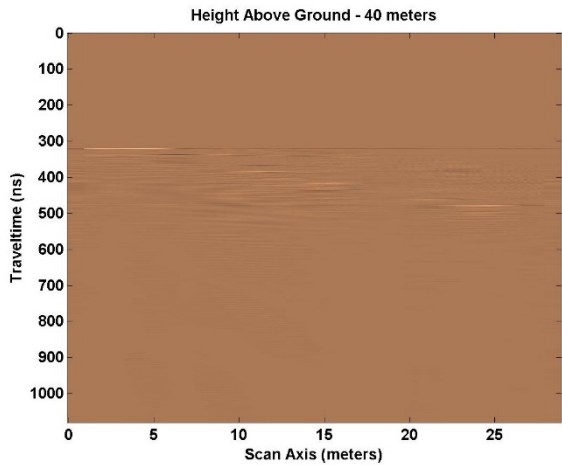


Figure 17. Output response of SA2, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

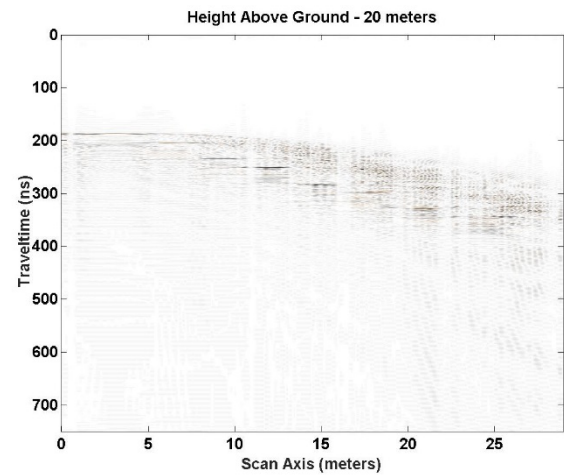


Figure 15. Output response of SA2, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

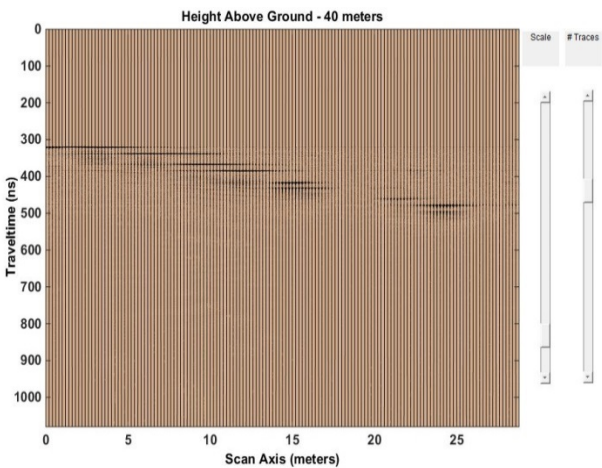


Figure 18. Signal traces of SA2 output response of EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

A third defined space model, Simulated Analysis 3 (SA3), was designed to examine the response of EM GMM algorithm on a defined space where the media is non-uniform. SA3 contains corrugated aluminum sheets modelled as perfect electrical conductors in dry sand, clay, granite, concrete and limestone media (Figure 19). The relative permittivity of each medium is noted in same figure. The details of the model are the same as SA2 except for the media used. Figure 19, Figure 21, Figure 23 and Figure 25 display the model with Tx's and Rx's at 4 different heights. Tx/Rx heights are 5, 10, 20, 40 meters above ground. GPR scanning frequencies are 6 total, 20, 30, 50, 100, 500, and 900 MHz. Figure 20, Figure 22, Figure 24, Figure 26 and Figure 27 depict the response to the GPR scans processed using the EM GMM process. The direct wave and ground bounce signals have been removed by subtraction. As the Tx/Rx height above ground increases, locating the 8 sheets in the image is less clear, but they can be found. At the 40 meter height, the best depiction of all 8 sheets is the display of individual EM processed GPR signal traces (Figure 27) rather than the image response.

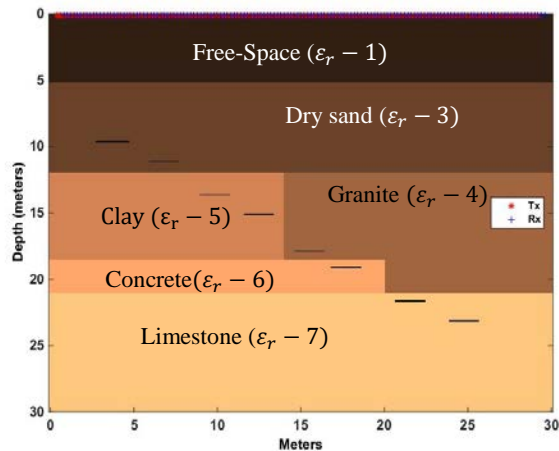


Figure 19. Defined Space of SA3, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 5 meters above ground.

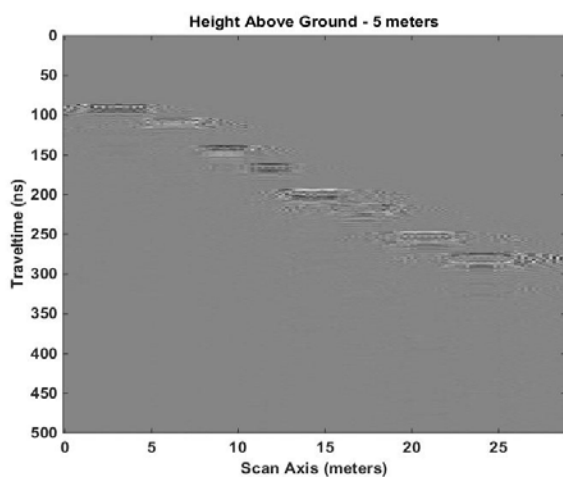


Figure 20. Output response of SA2, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

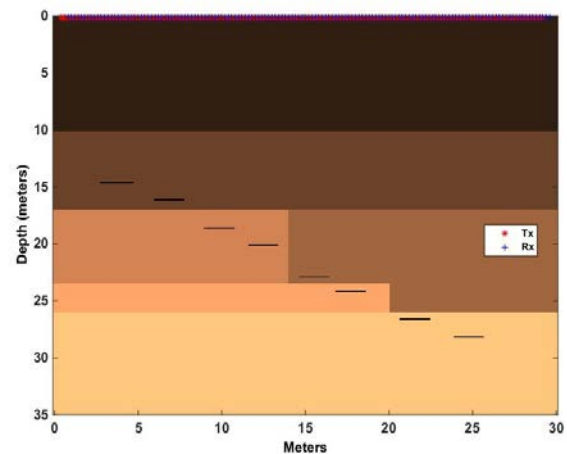


Figure 21. Defined Space of SA3, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 10 meters above ground

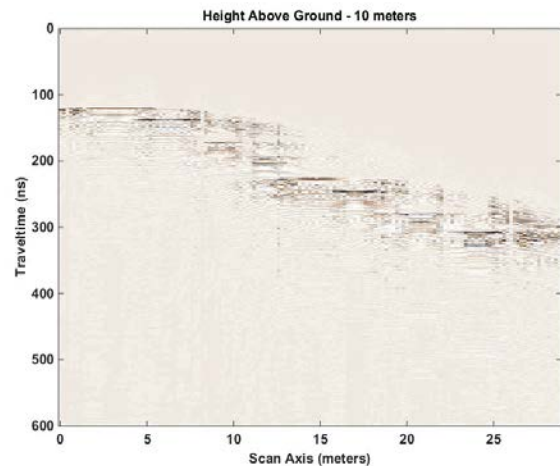


Figure 22. Output response of SA3, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

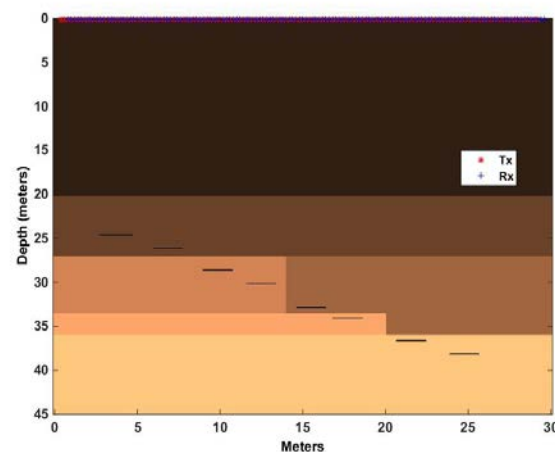


Figure 23. Defined Space of SA3, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 20 meters above ground.

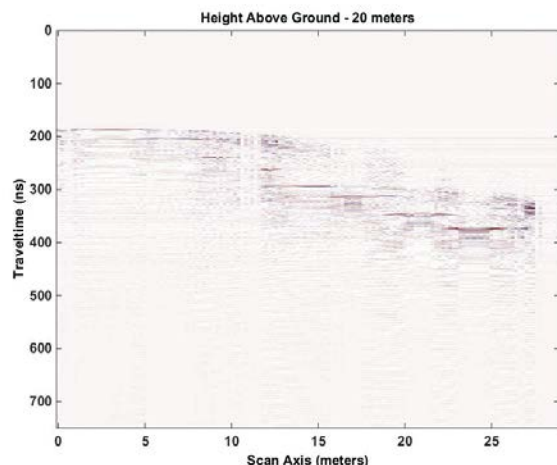


Figure 24. Output response of SA3, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

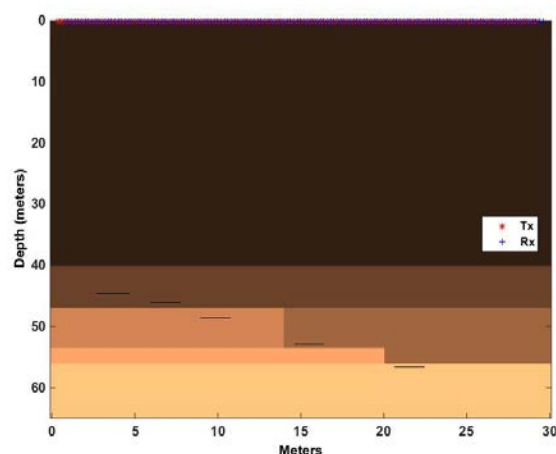


Figure 25. Defined Space of SA3, (8) 2 meter long plates, 0.1 meter thick with Tx's & Rx's 40 meters above ground

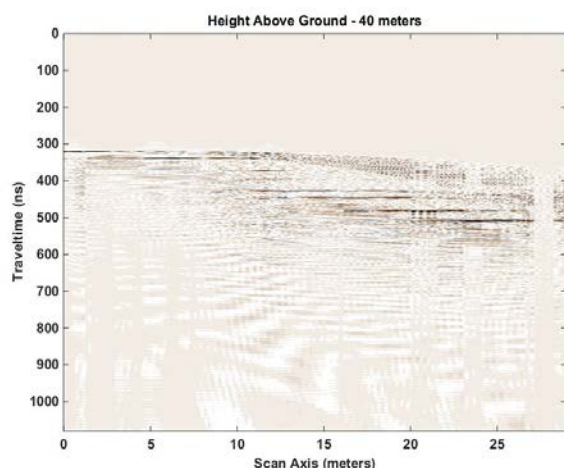


Figure 26. Output response of SA3, EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

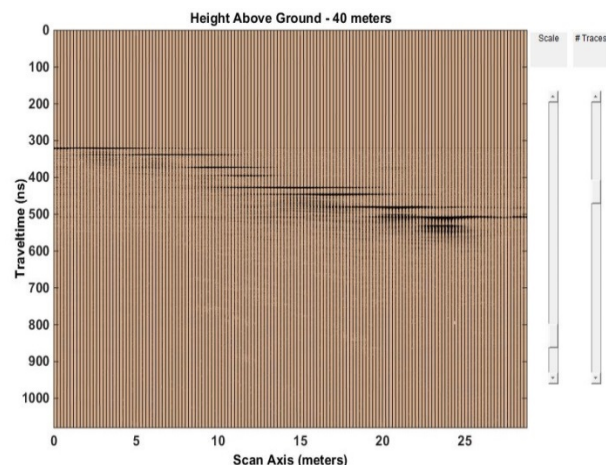


Figure 27. Signal traces of SA3 output response of EM sum of frequency signals with Direct Arrival and ground bounce signals removed.

VI. CONCLUSION AND FUTURE WORK

In this paper, as an extension of reference [1], we explored the use of the Expectation Maximization Gaussian Mixture Model method [3] to combine multiple frequency scans of the same target area. We examined the effectiveness of the EM GMM method on ground Penetrating Radar scans using transmitter and receiving antennas placed at various heights over the same target areas and media types as described in reference [1]. We conducted the analyses using the software program GprMax [5] due to the lack of actual hardware, real areas to scan and to compare results with reference [1] examples. Actual 3-D hardware verification of the GprMax software was determined in reference [19].

As part of the related work discussion, we reviewed the Maximum Likelihood Estimation process and its problem of working with hidden or incomplete data [3][4]. When hidden or incomplete data exists, a closed form solution of the MLE equation or a single global maximum is not easily obtained and very hard to solve for. We reviewed the EM GMM algorithm process and its benefit of working easily with hidden or incomplete data; creating a set of optimization problems simpler and guaranteed to converge while ultimately producing a global maximum after several iterations of producing increasing local maxima. We also, briefly reviewed other methods of compositing found in the literature. Methods of Dougherty et al. [7], Booth et al. [8][9] and Bancroft [10] we found to be less effective than our EM GMM method of reference [2]. Lastly, we found in the literature that scanning from various heights has been the purview of Synthetic Aperture Radars. SAR technology information obtained [11][12][13] did not concern itself with compositing but, did research methods to stitch area scans together, account for phase shifts [14][15] in the data, due to scanning angles, and distinguish objects with an author developed "Method of Moments Algorithm" [11].

Using GprMax [5], we repeated scanning the same test areas using the same 6 frequencies (20, 30, 50, 100, 500, and 900 MHz) as our previous work at 5, 10, 20 and 50 meters above ground. Our method performed well with outcomes similar to our previous results of scans with Tx's and Rx's near the ground. Images at heights above 20 meters were a challenge to recognize independent of the media that surrounding the targets. Moist sand, dry sand, concrete, clay granite and limestone did not change the end resulting image perceptibly. The method used to remove the direct wave/ground bounce signal also removed the boundary reflections for non-homogenous media. The test areas were scanned with the media and targets in place; then re-scanned with media in place without the targets. The scan with media and targets was subtracted from the scan with media without targets.

Problem areas remaining to be addressed are edge detection capability, removal of direct wave/ground bounce without removal of the reflected target responses, alignment of GPR trace starting points across frequencies and accounting for phase-shifts in the data. The SAR method solution was to use Gazdag [14] or F-K migration [15] techniques to manage phase-shifts in the data. We are encouraged that these methods will address the edge detection problem favorably.

ACKNOWLEDGMENT

This work was performed under the auspices of Sandia National Laboratories a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc. for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA-0003525.

REFERENCES

- [1] R. Tilley, H. Sadjadpour, and F. Dowla, "Combining Ground Penetrating Radar Scans of Differing Frequencies Through Signal Processing," The Ninth International Conference on Advanced Geographic Information Systems, Applications, and Services, GEOProcessing 2017, Nice, France, Mar 2017, pp. 32-38, ISBN:978-1-61208-539-5.
- [2] R. Tilley, H. Sadjadpour, F. Dowla, "Compositing Ground Penetrating Radar Scans of Differing Frequencies for Better Depth Perception", International Journal on Advances in Software, vol. 10, no. 3 & 4, year 2017, pp 413-431, ISSN 1942-2628. <https://www.iariajournals.org/software/tocv10n34.html>, 2018.7.30
- [3] A. P. Dempster, N.M. Laird and D.B. Rubin, "Maximum Likelihood from Incomplete Data via the EM Algorithm," Journal of the Royal Statistical society, Series B (Methodological) 39(1): pp. 1-38, 1977, JSTOR 2984875.MR0501537.
- [4] C. R. Shalizi, "Advanced Data Analysis from an Elementary Point of View," Book Draft from Lecture Notes for Course 36-402 at Carnegie Mellon University, Chapters 19.1-19.2.2, January 2017.
- <http://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/ADAfaEPoV.pdf>, 2017.11.23.
- [5] A. Giannopoulos, "Modelling Ground Penetrating Radar by GprMax," Construction and Building Materials, vol. 19, pp. 755-762, Dec 2005, DOI 10.1016/j.conbuildmat.2005.06.007.
- [6] J. A. Pena, and T. Teixido, "Cover Surfaces as a New Technique for 3-D Image Enhancement, Archaeological Applications," Repositorio Institucional de la Universidad de Granada, Spain, 2012, <http://hdl.handle.net/10481/22949>, 2017.11.23.
- [7] M. E. Dougherty, P. Michaels, J. R. Pelton, and L. M. Liberty, "Enhancement of Ground Penetrating Radar Data Through Signal Processing," Symposium on the Application of Geophysics to Engineering and Environmental Problems 1994, pp. 1021-1028, Jan 1994, DOI 10.4133/1.2922053.
- [8] A. L. Endres, A. Booth, and T. Murray, "Multiple Frequency Compositing of Spatially Coincident GPR Data Sets," Proceedings of the Tenth International Conference on Ground Penetrating Radar, 2004, Delft, The Netherlands, June 2004, pp. 271-274, ISBN: 90-9017959-3.
- [9] A. D. Booth, A. L. Endres, and T. Murray, "Spectral Bandwidth Enhancement of GPR Profiling Data Using Multiple-Frequency Compositing," Journal of Applied Geophysics, vol 67, pp. 88-97, Jan 2009, DOI 10.1016/j.jappgeo.2008.09.015.
- [10] S. W. Bancroft, "Optimizing the Imaging of Multiple Frequency GPR Datasets using composite Radargrams: An Example from Santa Rosa Island, Florida," PhD dissertation, University of South Florida, 2010.
- [11] S. Vitebskiy, L. Carin, and M. Ressler, "Ultra-Wideband, Short-Pulse Ground-Penetrating Radar: Simulation and Measurement," IEEE Transactions on Geoscience and Remote Sensing, Vol. 35, NO. 3, May 1997, pp. 762-772. <https://www.math.ucdavis.edu/~saito/data/sonar/vitebskiy.pdf>, 2018.7.30.
- [12] M. Skjelvareid, "Synthetic aperture ultrasound imaging with application to interior pipe inspection", PhD dissertation, University of Tromso, 2012. ISBN 978-82-8236-067-8, <http://hdl.handle.net/10037/4649>, 2018.7.30.
- [13] H. Zhang, W. Benedix, D. Plettemeier, and V. Ciarletti, "Radar Subsurface Imaging by Phase Shift Migration Algorithm," 2013 European Microwave Conference, Nuremberg, 2013, pp. 1843-1846. DOI: 10.23919/EuMC.2013.6687039. <https://ieeexplore.ieee.org/iel7/6679726/6686544/06687039.pdf>, 2018.7.30.
- [14] J. Gazdag, "Wave Equation with the phase-shift method", Geophysics, Vol. 43, NO. 7, December 1978, pp. 1342-1351. DOI: 10.1190/1.1440899. <https://doi.org/10.1190/1.1440899>, 2018.7.30
- [15] R. Stolt, "Migration by Fourier Transform", Geophysics, Vol. 43, NO. 1, February 1978, pp 23-48. DOI: 10.1190/1.1440826. <https://www.math.ucdavis.edu/~saito/data/sonar/stolt.pdf>, 2018.7.30.
- [16] Padhraic Smyth, "The EM Algorithm for Gaussian Mixtures, Probabilistic Learning: Theory and Algorithms, CS274A," University of California, Irvine, Department of Computer Science, Lecture Note 4.
- [17] J. J. Verbeek, N. Vlassis, and B. Kröse, "Efficient Greedy Learning of Gaussian Mixtures," The 13th Belgian-Dutch Conference on Artificial Intelligence (BNAIC'01), pp. 251-258, 2001, INRIA-00321510.
- [18] A. Tavlove, "Review of the formulation and Applications of the Finite-Difference Time-Domain Method for Numerical Modeling of Electromagnetic-Wave Interactions with Arbitrary Structures," Wave Motion, vol. 10, pp. 547-582, Dec 1988, DOI 10.1016/0165-2125(88)90012-1.

- [19] R. Tilley, F. Dowla, F. Nekoogar, and H. Sadjadpour, "GPR Imaging for Deeply Buried Objects: A Comparative Study Based on FDTD models and Field Experiments", Selected Papers Presented at the MODSIM World 2011 Conference and Expo; pp. 45-51, 2012; (NASA/CP-2012-217326); (SEE 20130008625).
<https://ntrs.nasa.gov/archive/nasa/casi.ntrs.nasa.gov/20130008669.pdf>, 2018.11.13.

Earth Observation Semantics and Data Analytics for Coastal Environmental Areas

Corneliu Octavian Dumitru, Gottfried Schwarz, and Mihai Datcu

Remote Sensing Technology Institute, German Aerospace Center (DLR), 82234 Wessling, Germany
corneliu.dumitru@dlr.de, gottfried.schwarz@dlr.de, mihai.datcu@dlr.de

Abstract — Current satellite images provide us with detailed information about the state of our planet, as well as about our technical infrastructure and human activities. A range of already existing commercial and scientific applications try to analyze the physical content and meaning of satellite images by exploiting the data of individual, multiple or temporal sequences of images. However, what we still need today are advanced tools to automatically analyze satellite images in order to extract and understand their full content and meaning. To remedy this exploration problem, we outline a highly automated and application-adapted data-mining and content interpretation system consisting of five main components, namely Data Sources (selection and storage of relevant images), Data Model Generation (patch cutting and generation of feature vectors), Database Management System (systematic data storage), Knowledge Discovery in Databases (clustering and content labeling), and Statistical Analytics (generation of classification maps). As test sites, we selected UNESCO-protected areas in Europe that include coastal areas for monitoring and an area known in the Mediterranean Sea that contains fish cages. The analyzed areas are: the Curonian Lagoon in Lithuania and Russia, the Danube Delta in Romania, the Hardangervidda in Norway, and the Wadden Sea in the Netherlands. For these areas, we are providing the results of our image content classification system consisting of image classification maps and additional statistical analytics based on three different use cases. The first use case is the detection of wind turbines vs. boats in the Wadden Sea. The second use case is the identification of fish cages/aquaculture along the Mediterranean coast. Finally, the third use case describes the differences between beaches, dams, dunes, and tidal flats in the Danube Delta, the Wadden Sea, etc. The average classification accuracy that we obtained is ranging from 80% to 95% depending on the type of available images.

Keywords - coastal monitoring; data mining; protected areas; Sentinel-1; Sentinel-2; TerraSAR-X.

I. INTRODUCTION

In Earth observation (EO) [1], a very popular satellite image analysis system is the one from Digital Globe, named Tomnod, or Google Earth together with its related tools, which are targeting general user topics. In addition, in the EO domain, there are systems such as LandEX [2], which is a land cover analysis system, while GeoIRIS [3] is a system that allows the user to refine a given query by iteratively specifying a set of relevant, and a set of non-relevant images. A similar information retrieval system is IKONA [4], which is using relevance feedback in order to exploit very high resolution EO images. Further, the Knowledge-driven

Information Mining (KIM) system [5] is an example of an active learning system providing semantic interpretation of image content. The KIM concept evolved into the TELEIOS prototype [6], complementing the scope of searching for EO images with additional geo-information and in-situ data integrated into an operational EO system [7] to interpret TerraSAR-X images. A similar concept to the KIM concept is presented in [8], while in [9] a data mining approach for Big Data is described.

Our proposed system is very fast compared with the other existing systems, and can retrieve with only a few examples the desired category with higher accuracy. The diversity of applications that can be considered for such systems are rather broad and include, for instance, coastal environmental monitoring (sea level, tides and wave direction), land cover/use changes, disaster monitoring, forest management, ice monitoring, monitoring of active volcanoes, waste deposit site management, traffic monitoring, vegetation monitoring, urban sprawl, soil moisture dynamics, etc.

The paper is organized as follows. Section II describes the selected test areas with a number of results obtained with the proposed methodology. Section III presents our datasets. Section IV details the data mining methodology applied in this paper. Section V shows the results for three selected cases. Section VI concludes the paper together with the future work. The acknowledgements close the paper.

II. SELECTION OF TEST AREAS, USE CASES, AND APPLICATIONS

The use case selection is closely related to the ECOPOTENTIAL project that focuses on a targeted set of internationally recognized protected areas in Europe, European territories and beyond, including mountainous, arid and semi-arid, and coastal and marine ecosystems [10].

We emphasize here a number of use cases for monitoring coastal environments. The first four use cases are internationally recognized protected areas as UNESCO (United Nations Educational, Scientific and Cultural Organization) Natural Heritage sites. These selected use cases are the Wadden Sea with the Dutch Delta (in the Netherlands), the Danube Delta (in Romania), the Curonian Lagoon (in Lithuania and Russia), and the Hardangervidda (in Norway). The last use case is an area that has a large aquaculture located between Albania and Greece.

A. The Wadden Sea, Netherlands

Site description: The Wadden Sea (Dutch: Waddenzee, German: Wattenmeer, Danish: Vadehavet) is an intertidal

zone in the south-eastern part of the North Sea. It lies between the coast of N-W continental Europe and the range of Frisian Islands, forming a shallow body of water with tidal flats and wetlands [11], protected by a 450 km long chain of barrier islands, the Wadden Islands. The Wadden Sea region measures about 22,000 km², divided between land and sea. About 63% of the region lies in Germany, with about 30% in the Netherlands, and 7% in Denmark [12]. In 2009, the Dutch-German Wadden Sea was inscribed on the UNESCO World Heritage List, and the Danish part was added in 2014.

The landforms in the Wadden Sea region have essentially been created from a marine or tidal environment [13].

Typical for the Wadden Sea are large tidal flats, which are characterized by very high benthic biomass and productivity, dominated by molluscs and polychaetes.

State-of-the-art publications: In the research literature there are several studies treating the Wadden Sea area along the years. In order to understand the Wadden Sea dynamics, a number of recent publications [14][15][16][17] already used remote sensing images and addressed the issue of Synthetic Aperture Radar (SAR) satellite image classification and interpretation in these areas. At present, the option of data fusion from different sensors has not yet been fully exploited.

Image interpretation goal: The Wadden Sea area faces a strong economic impact due to recreation, fisheries and maritime traffic. The last impact is due to, e.g., the ports of Bremerhaven, Hamburg, and Rotterdam whereby the traffic runs through or nearby this area, which makes the monitoring of sand banks and any decrease of the water depth and the tide levels in this area a critical topic for maritime security. A second important topic is the monitoring of biodiversity as described by [10].

Typical examples: The diversity of categories identified from a single image and a typical classification map of the Wadden Sea and its surrounding areas are shown in Figures 1 and 2.

B. The Danube Delta, Romania

Site description: The Danube Delta is the second largest river delta in Europe and is the best preserved one on the continent [18]. Formed over a period of more than 10,000 years, the Danube Delta continues to grow due to the 67 million tons of alluvia deposited every year by the Danube River [19]. The delta is an ideal test and validation area for vegetation monitoring as it is characterized by high biodiversity and various crops.

The Delta is formed around the three main channels of the Danube, named after their respective ports Chilia (in the north), Sulina (in the middle), and Sfântu Gheorghe (in the south).

The greater part of the Danube Delta lies in Romania (Tulcea County), while its northern part, on the left bank of the Chilia arm, is situated in Ukraine (Odessa Oblast). Its total surface is 4,152 km² of which 3,446 km² are in

Romania. The waters of the Danube, which flow into the Black Sea, form the largest and best preserved delta in Europe. In 1991, the Danube Delta was inscribed on the UNESCO World Heritage List due to its biological uniqueness.

State-of-the-art publications: In the image processing literature there are not many studies treating the Danube Delta especially for SAR data [20][21][22]. However, the monitoring of biodiversity from in-situ measurements has attracted more interest [23].

Image interpretation goal: At the mouth of the Danube, the alluvial discharge decreases every year from 81 million tons in 1894, to 70 million tons in 1939, 58 million tons in 1982, and about 22 million tons in 2015. This makes it interesting to monitor the evolution of the alluvial discharge and to investigate its impact on the Danube Delta and the three channels together with their ports (Chilia, Sulina, and Sfântu Gheorghe) through the years.

The data can be combined with other types of information, such as the volume of water of each channel in order to prepare flood risk maps needed for the safety of the shipping traffic and also for the local authorities to protect the human settlements. Another image interpretation goal is vegetation monitoring, in particular, biodiversity issues and crop type analyses.

Typical examples: The diversity of categories identified from a single image and a typical classification map of the Danube Delta and its surrounding areas are shown in Figures 3 and 4.

C. The Curonian Lagoon, Lithuania and Russia

Site description: The Curonian Lagoon is the largest European lagoon. Situated in the southern part of the Baltic Sea with a total area of 1584 km², the lagoon receives water from the River Nemunas. The salinity of the water is higher and fluctuates between the northern and southern part of the lagoon [10]. The entire Lithuanian part of the Curonian Lagoon has been designated as a NATURA 2000 area and in 2000 the Curonian Spit cultural landscape was as well inscribed on the UNESCO World Heritage List.

State-of-the-art publications: In the remote sensing literature, there are not many studies treating the Curonian Lagoon especially for SAR data. However, the monitoring of biodiversity has attracted greater interest [24][25][26].

Image interpretation goal: We analyzed the effect of socio-economic activities of the area regarding: the ceasing commercial fisheries, the prohibition of the extraction of mineral resources, the agricultural sector, the hunting sector, the restriction of recreational use of the aquatic areas, and the oil drilling/pollution of the area.

Typical examples: The diversity of categories identified from a single image and a typical classification map of the Curonian Lagoon and its surrounding areas are shown in Figures 5 and 6.

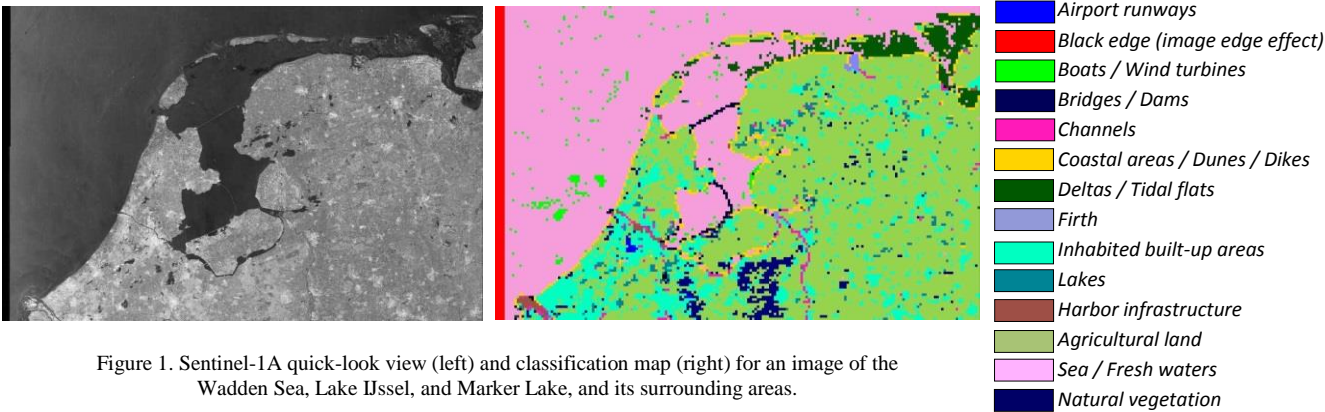


Figure 1. Sentinel-1A quick-look view (left) and classification map (right) for an image of the Wadden Sea, Lake IJssel, and Marker Lake, and its surrounding areas.

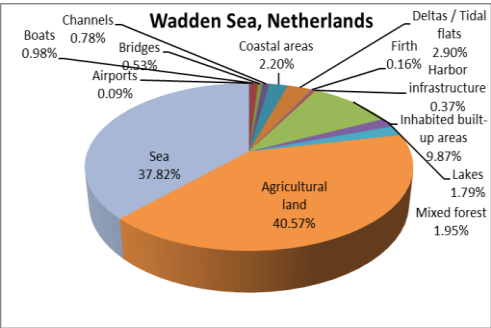


Figure 2. Diversity of categories identified from a single image of the Wadden Sea, the Netherlands.

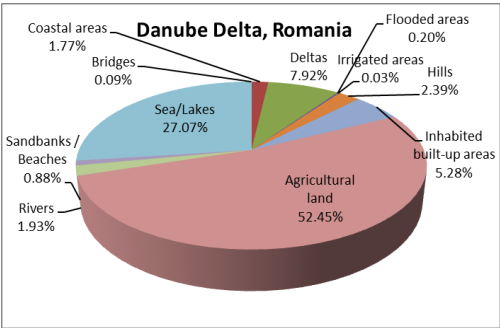


Figure 4. Diversity of categories identified from a single image of the Danube Delta.

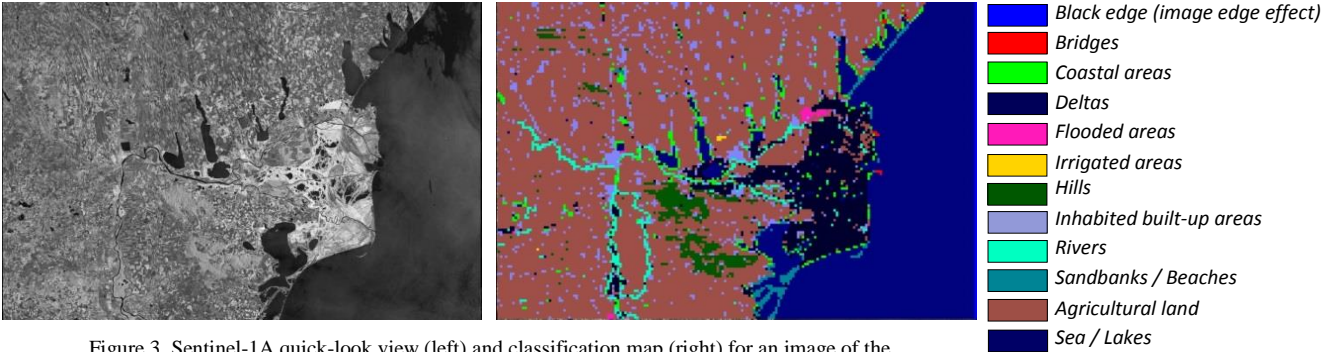


Figure 3. Sentinel-1A quick-look view (left) and classification map (right) for an image of the Danube Delta and the surrounding areas.

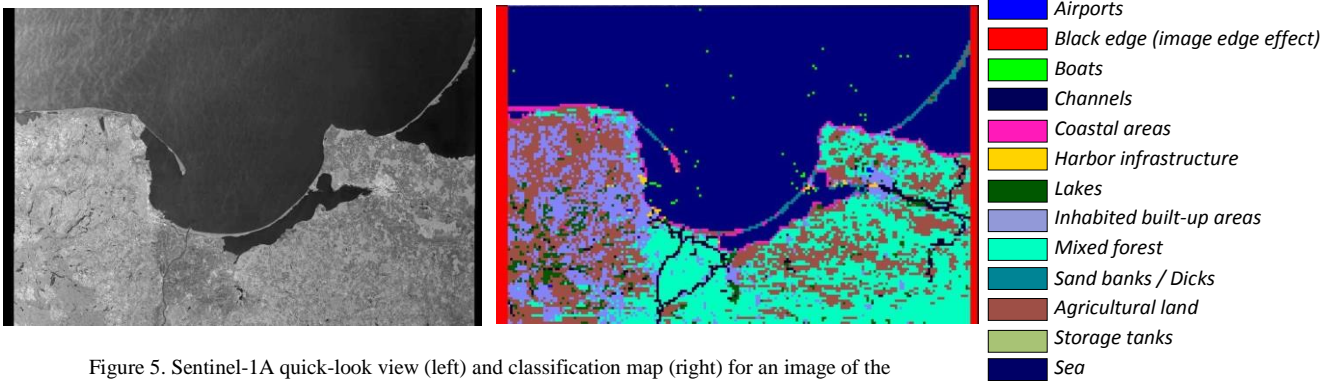


Figure 5. Sentinel-1A quick-look view (left) and classification map (right) for an image of the Curonian Lagoon and the surrounding areas.

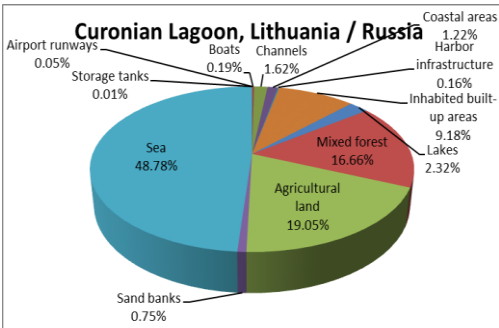


Figure 6. Diversity of categories identified from a single image of the Curonian Lagoon.

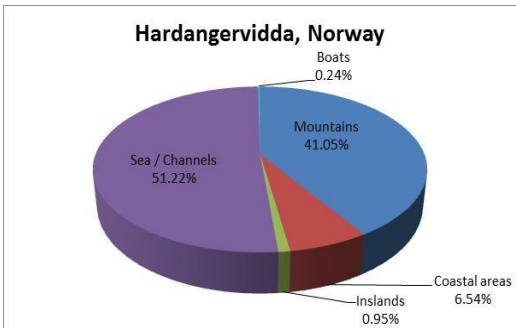


Figure 8. Diversity of categories identified from single images of the Hardangervidda.

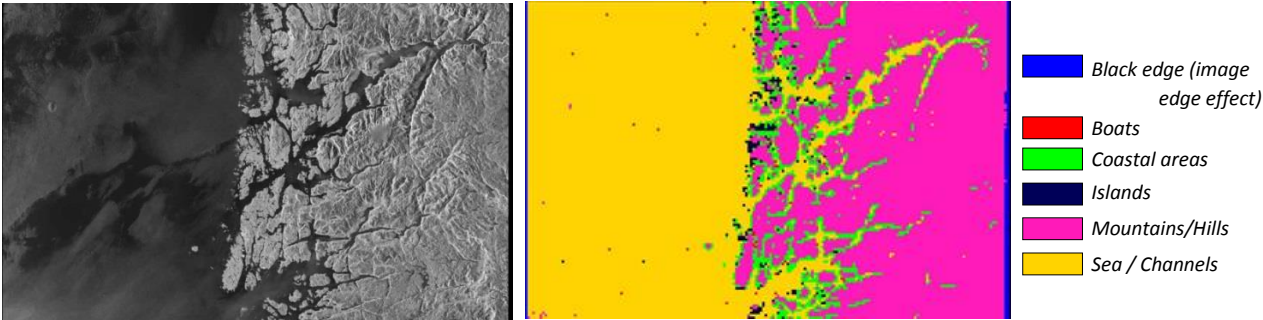


Figure 7. Sentinel-1A quick-look view (left) and the "patch-based" classification map (right) for an image of the Hardangervidda, Norway.

D. The Hardangervidda, Norway

Site description: The Hardangervidda is the largest peneplain (eroded plain) plateau in Europe with a cold year-round alpine climate, and one of Norway's largest glaciers. The largest extent covers an area of about 6,500 km² at an average elevation of 1100 m and is part of the Hardangervidda National Park, which is a protected area. The plateau with its boulders and rock outcrops are the remnants of mountains that were worn down by the glaciers during the quaternary Ice Ages [10]. The landscape of the Hardangervidda is characterized by barren, treeless, and shrubby moorland interrupted by numerous lakes, streams and rivers.

State-of-the-art publications: In the remote sensing literature, there are very few publications that are

investigating the Hardangervidda and its surrounding areas using SAR data. One of these publications is the monitoring of wet snow [27].

Image interpretation goal: We analyzed the effects in time of the high mountain plateau with its unique arctic flora and fauna, and the land use (e.g., grazing by livestock and fishing) [10].

Typical examples: The diversity of categories identified from a single image and a typical classification map of the Hardangervidda and its surrounding areas are shown in Figures 7 and 8.

E. Aquaculture, Albania and Greece

Site description: Along the Mediterranean coast we can find a number of aquaculture areas/fish cages.

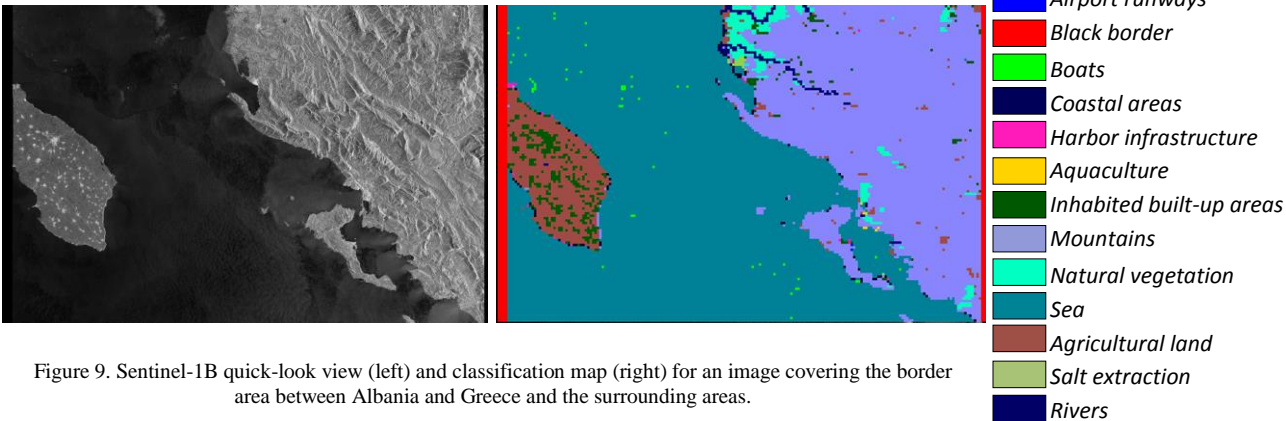


Figure 9. Sentinel-1B quick-look view (left) and classification map (right) for an image covering the border area between Albania and Greece and the surrounding areas.

The authors of [28] identified 248 cages with a circular diameter larger than 40 m and 20,976 cages within 10 km offshore. The majority of these fish cages/aquaculture (see Figure 1 in [29]) are located in Greece (49%) and Turkey (31%). Based on this context, we selected an area along the Greek coast near Corfu Island.

State-of-the-art publications: In this area of aquaculture / fish cages, there are many remote sensing publications. In two recent publications ([30] and [31]), the authors are using Sentinel-1 SAR images in order to monitor aquaculture and to count fish cages.

Image interpretation goal: Using the satellite imagery available through the Sentinels or TerraSAR-X, we can detect these fish cages/aquaculture with a better or lower accuracy using SAR or optical data. This information can be used later to estimate the fish farm production in the area. We can extend the analysis to the entire Mediterranean coast and the results can be compared with the reports [32] published by FAO (United Nations Food and Agriculture Organization).

Typical examples: The diversity of categories identified from a single image and a typical classification map of the area between Albania and Greece close to Corfu Island are shown in Figures 9 and 10.

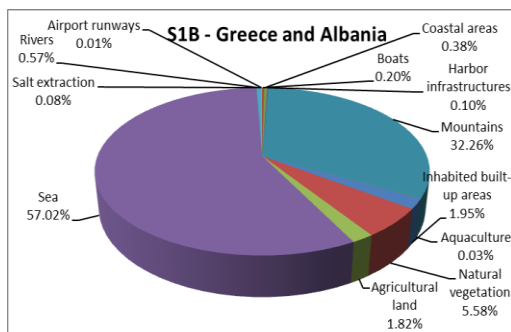


Figure 10. Diversity of categories identified from a single image of the area between Albania and Greece and its surrounding areas.

III. DATASETS

An important aspect to be addressed is the compilation of a reference dataset for test and validation of the different systems. We already possess an initial synthetic aperture radar dataset composed of 1000 TerraSAR-X images and 100 Sentinel-1 images covering target areas from around the world.

From this database, about 295 TerraSAR-X and 25 Sentinel-1A images have already been annotated by a remote sensing expert using a semi-automatic semantic annotation system resulting in a semantic catalogue of hundreds of semantic labels grouped in a 3-level hierarchical scheme [33]. This annotated database mainly covers urban and industrial areas together with their infrastructure predominantly from Europe, and can be considered as our initial ground truth dataset [34].

Our latest dataset also contains optical satellite data with multi-spectral images (e.g., Sentinel-2A), and synthetic aperture radar images (e.g., TerraSAR-X and Sentinel-1A / 1B). These data cover 10 protected areas from Europe (national parks, mountains, arid and semi-arid areas, and coastal and marine ecosystems) [10].

IV. METHODOLOGY

The data mining system [7] (used in this paper) is composed of five modules: Data Sources (DS), Data Model Generation (DMG), Database Management System (DBMS), Knowledge Discovery in Databases (KDD), and Statistical Analytics (SA). It's the first one that explores, discovers, extracts semantics, and understands Big EO data.

Our data mining methodology is shown in Figure 11 and a pseudo-code segment is given in Table 1. For more details about the implementation and algorithms, see [7].

The DS module collects the relevant data related to each use case to be processed and analyzed in the other modules.

The DMG module transforms the original format of original Earth observation products into smaller and more compact product representations that include image descriptors, metadata, image patches, etc.

The DBMS module is used for storing all the generated information and allows querying and retrieval of the available image data.

The KDD module is in charge of finding patterns of interest from the processed data and presenting them to the user. Moreover, the KDD module allows annotating the image content by using machine learning algorithms and human interaction resulting in physical categories.

The SA module provides classification maps of each dataset and distribution results of the retrieved categories in an image.

These five modules are operated automatically and interactively with and without user interaction.

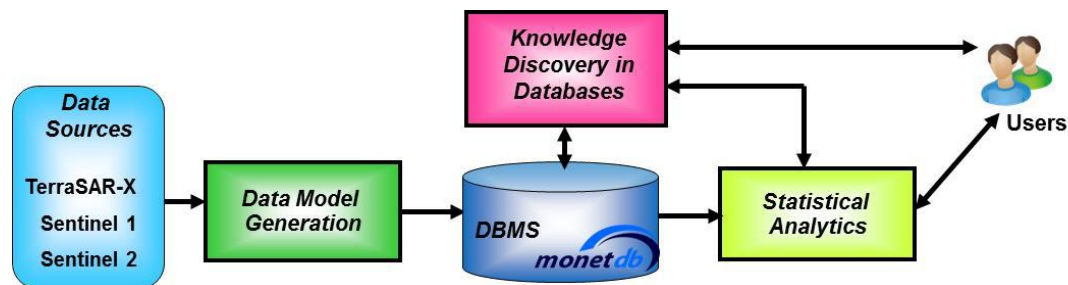


Figure 11. The proposed data mining system that includes the following modules: Data Sources, Data Model Generation, Database Management System, Knowledge Discovery in Databases (KDD), and Statistical Analytics (SA) [6].

TABLE I. THE PROPOSED METHODOLOGY.

```

Step 1: Data Sources (DS) EO Dataset
Select and download typical EO images and store
them into our EO Dataset.

Step 2: Data Model Generation (DMG)
for each  $EO_i$  image ( $i=1 \dots N$ ) do
    tile  $EO_i$  into  $p_{i,j}$  patches ( $j=1 \dots M$ ), where the
    size of the patches depends on the image
    resolution
    store all  $p_{i,j}$  into the DBMS
    for each  $p_{i,j}$  patch do
        extract an  $f_{i,j}$  primitive feature vector
        from optical / SAR algorithms
        //e.g., Gabor filters with 5 scales and 6
        orientations and compute the means and
        standard deviations of the coefficients //
        store all  $f_{i,j}$  vectors into the DBMS
    end
end

Step 3: Knowledge Discovery in Databases (KDD)
if  $r_k$  ( $k=1 \dots K$ )  $\nexists$  do //if the patch reference label
    has not yet been generated//
    for all  $f_{i,j}$  primitive feature vectors do
        group the  $f_{i,j}$  into  $g_k$  clusters and group
        them into  $c_k$  categories using an SVM
        (Support Vector Machine)
        for each  $c_k$  category do
            select an  $r_k$  semantic annotation label
            //visual support via Google Earth //
            store reference  $r_k$  labels into the DBMS
        end
    end
else // routine processing after label generation//
    for all  $f_{i,j}$  primitive feature vectors do
        group the  $f_{i,j}$  into  $g_l$  clusters ( $l=1 \dots L$ ) and
        group them into  $c_l$  categories using an
        SVM
        store all  $g_l$  into the DBMS
        for each  $c_l$  category do
            select an  $a_l$  semantic annotation
            //visual support via Google Earth//
            store  $a_l$  labels into the DBMS
        end
    end
end

Step 4: Statistical Analytics (SA)
for selected  $EO_i$  and its  $a_l$  do
    generate classification maps
    compare obtained  $a_l$  annotations with  $r_k$ 
    //reference annotations (generated previously)//
    and generate change maps
    compute characteristic metrics
    //e.g., precision/recall by comparing the results
    with the  $r_k$  //
end

```

V. RESULTS AND DISCUSSIONS

For the selected areas of interest, different use cases can be considered such as: detection of *Wind turbines vs. Boats*; *Fish cages/Aquaculture*; differences between *Beaches, Tidal flats*, and *Dams*; etc.

For our first example, we selected the Wadden Sea area and we show the results for the detection of *Wind Turbines vs. detection of Boats*. The images were acquired in order to cover, as much as possible, the same area on the surface and/or the same acquisition date or a close date between the acquisitions. The data set consists of different images acquired by three different satellites: a TerraSAR-X image acquired on May 13, 2015 with a resolution of 2.9 meters, a Sentinel-1A image acquired on May 15, 2015 with a resolution of 20 meters, and a Sentinel-2A single quadrant-image acquired on April 21, 2016 with a resolution of 10 meters (comprising only the RGB bands). In Figure 12, we show the available data for the Wadden Sea protected area.

All these images were tiled into patches, and from each patch a feature vector was extracted. We classified the images considering only two categories of interest, namely *Wind turbines* and *Boats* (see Figure 13). Based on the extracted features and the specific patterns of these categories, we were able to separate them during classification. Figures 14, 15, and 16 illustrate the retrieved categories after the classification back-projected on the quick-look of each image product. For each image product, the locations of *Wind turbines* and *Boats* are marked in green and blue, respectively.

The complete processing chain from ingestion to annotation was run on a desktop PC with software coded in Java 8 and Matlab R2105a. The PC used for our experiments had a processor clock rate of 2.40 GHz, and a RAM capacity of 8 GB. Typically, we obtain a CPU usage of less than 25% as we store all image files onto disk and have to wait for the completion of all data transfers. The actual memory allocation of our PC configuration is less than 50 MBytes per image. The classification and display of a new set of retrieved patches needs about 4 to 6 ms when we have a collection volume of 2 GBytes of image data.

The accuracy of the results was computed separately for each sensor and for each retrieved category. For each image (EO_i) we compared the category a_l with its corresponding reference category r_k and we computed its classification accuracy. The attained average accuracy is 93%, ranging from 80% to 95% depending on the image type (e.g., TerraSAR-X, Sentinel-1A, or Sentinel-2A). When we compare the different SAR sensors, we notice that the overall classification accuracy is higher for the high resolution instruments, for example for TerraSAR-X.

For the second example, we selected an area between Albania and Greece in order to identify *Fish cages/Aquaculture* (Figure 17). The goal of this second example is to extract, from different image products, as much as possible, all the visible patches from the images that contain this category.

The location of the area of interest (*Fish cages/Aquaculture*) is close to Corfu Island and is shown in

Figure 18 and marked with pushpins [29]. The satellite images were acquired by different sensors around the same period: the TerraSAR-X product was acquired on January 19, 2017, the Sentinel-1B product on January 15, 2017, and the Sentinel-2A product on October 15, 2016 (when the area of interest was not covered by *Clouds*). Figure 19 shows the available data for the investigated area.

All three images are covering the area of interest where the *Fish cages/Aquaculture* are located, but not in all three images we can see the entire aquaculture. This is because of the sensor type (SAR or multi-spectral), resolution, and maybe because of the feature extraction methods being used for classification.

The images were tiled into patches, and from each patch a feature vector was extracted based on the type of sensor. Further, we classified the image patches and chose three semantic labels, namely *Aquaculture*, *Boats*, and *Harbor infrastructures*.

Figure 20 illustrates the retrieved categories after the classification projected on the quick-look image of each product. In each image the location of the semantic labels are marked in blue, red, and green.



Figure 12. Locations of the Wadden Sea shown on OpenStreetMap; the TerraSAR-X footprints are in green, the Sentinel-1A footprint is in orange, and the Sentinel-2A footprint (all quadrants) is in blue.

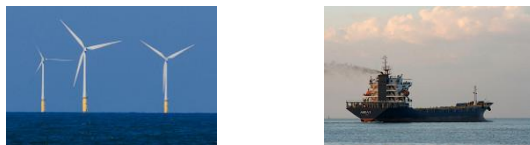


Figure 13. In-situ data: *Wind turbines* [35] vs. *Boats/Ships* [36].

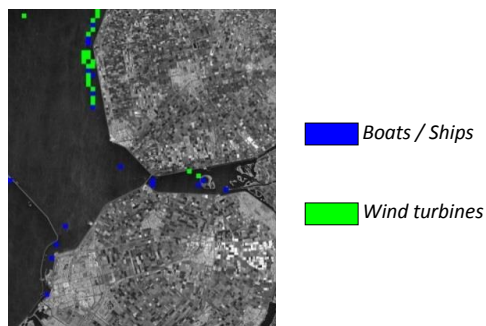


Figure 14. TerraSAR-X “patch-based” classification results of two categories back-projected onto a SAR image of Flevoland, the Netherlands.

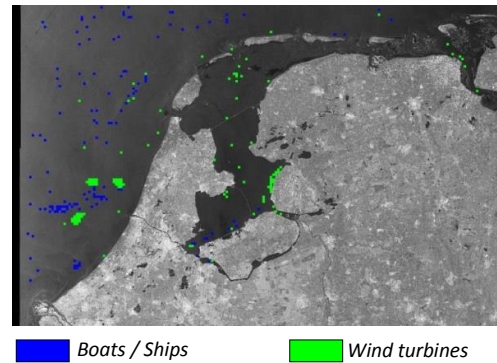


Figure 15. Sentinel-1A “patch-based” classification results of two categories back-projected onto a SAR image of the Wadden Sea, Lake IJssel, and Marker Lake, and the surrounding areas in the Netherlands.

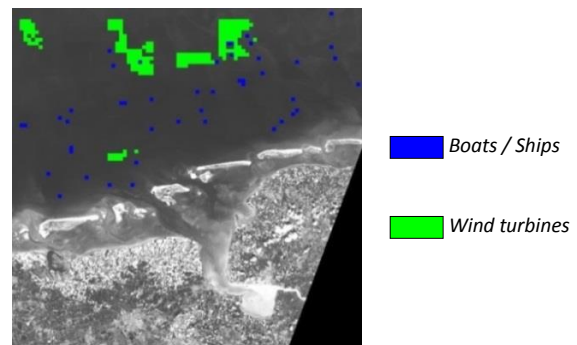


Figure 16. Sentinel-2A “patch-based” classification results of two categories projected on a (gray level) image of the German and Dutch Wadden Sea.



Figure 17. In-situ data: *Fish cages* [37] vs. *Aquaculture* [38].

We can conclude that, when the product has a higher resolution, we can better identify the area of interest, and with higher accuracy.

The third example aims at the differences between *Beaches*, *Dams*, *Dunes*, and *Tidal flats*, categories that we can find within the images available in our dataset (e.g., the Danube Delta, the Curonian Lagoon, or the Wadden Sea). In this case, we obtained similar accuracy results with the previous cases.

For illustration, we selected the area of the Wadden Sea and the area of the Danube Delta. In Figure 22, we show the results for the identification of *Dams*, *Dunes*, and *Deltas* or *Tidal flats* in the Wadden Sea. The image was acquired by Sentinel-1 on May 15, 2015 with a resolution of 20 meters. We tiled this image into patches and extracted a feature vector that is used further for classification. We classified the images considering only three categories of interest, namely *Dams*, *Dunes*, and *Tidal flats* (see the in-situ data in Figure 21). Based on the extracted features and the specific patterns of these categories, we were able to separate them during classification.

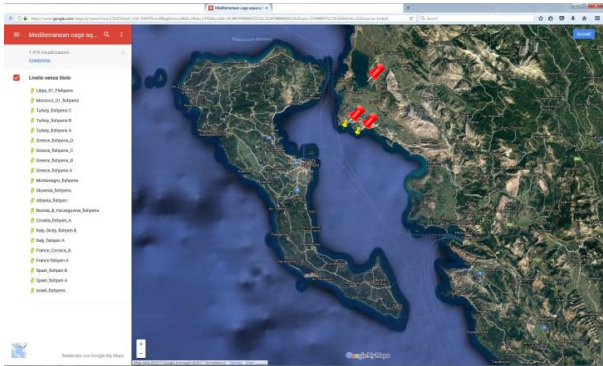


Figure 18. Locations of areas of interest (*Fish cages/ Aquaculture*) on Google Maps between Greece and Albania marked with red pushpins [29].

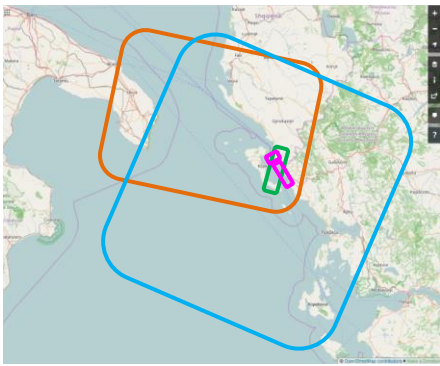


Figure 19. Locations of areas of interest (*Fish cages / Aquaculture*) on OpenStreetMap between Greece and Albania marked in purple. The satellite images were acquired by around the same date and by TerraSAR-X (in green), by Sentinel-1B (in orange), and by Sentinel-2A (in blue).

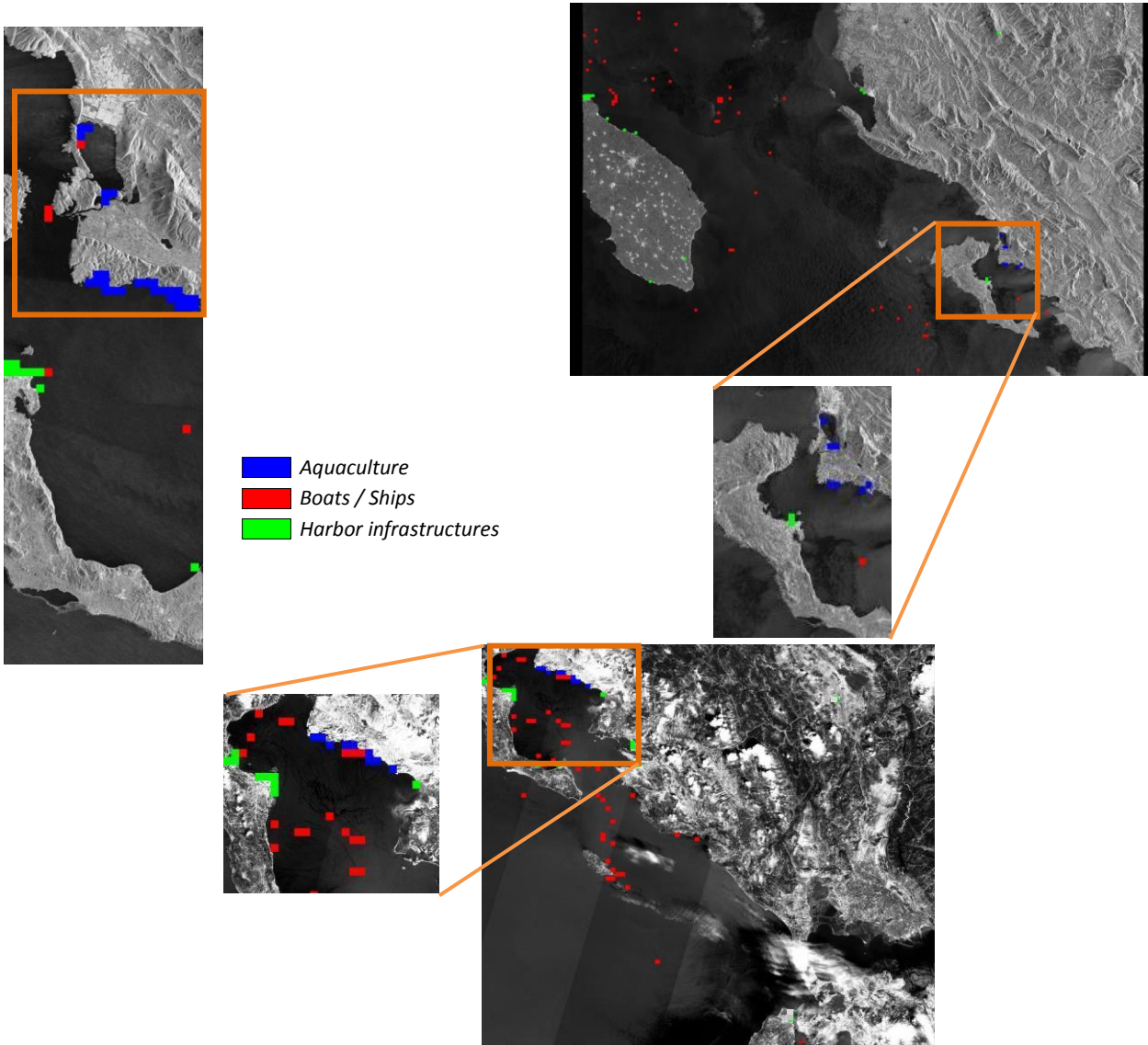


Figure 20. Comparative “patch-based” classification results of TerraSAR-X (top left), Sentinel-1B (top right), and (gray level) Sentinel-2A (bottom center) projected on an image covering our area of interest.

The classification map was generated by back-projecting it on the image quick-look of the retrieved categories; then the locations of *Dams*, *Dunes*, and *Tidal flats* were marked in blue, red, and green, respectively.

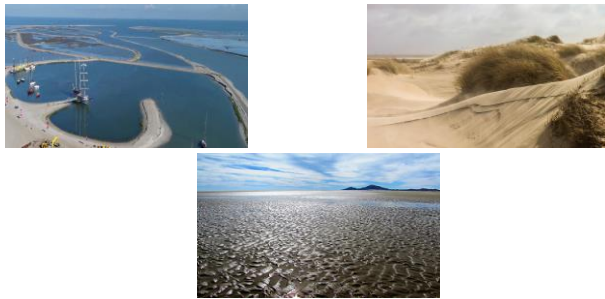


Figure 21. In-situ data: *Dams* [39] vs. *Dunes* [40] vs. *Tidal flats* [41].

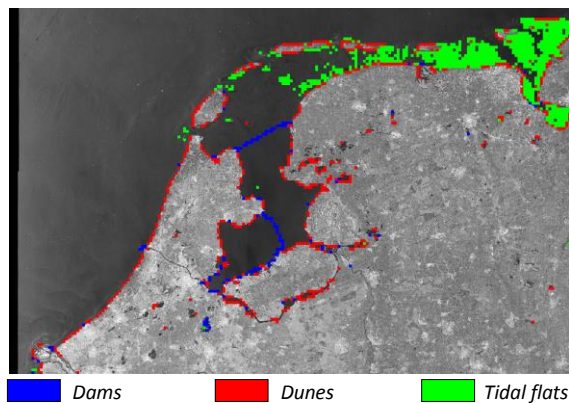


Figure 22. Sentinel-1A “patch-based” classification results of three categories back-projected onto a SAR image of the Wadden Sea, Lake IJssel, and Marker Lake, and the surrounding areas in the Netherlands.

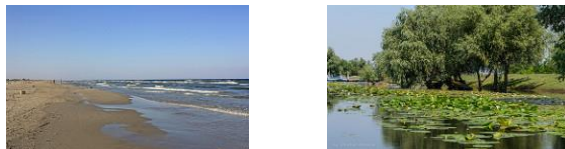


Figure 23. In-situ data: *Beaches* [42] vs. *Deltas* [43].

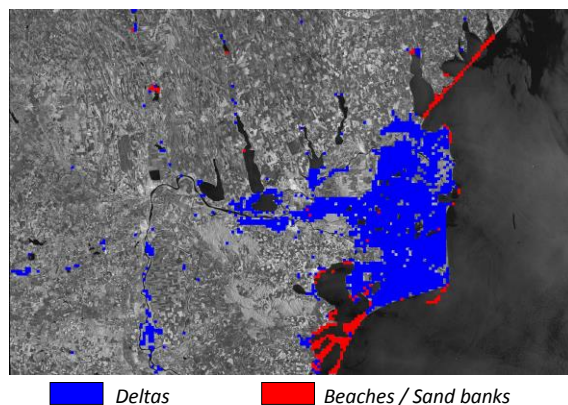


Figure 24. Sentinel-1A “patch-based” classification results of two categories back-projected onto a SAR image of the Danube Delta and its surrounding areas in Romania.

In Figure 24, we show the results for the identification of *Beaches* or *Sand banks* and *Deltas* in the Danube Delta. The classification was made by considering only two categories (see the in-situ data in Figure 23). Based on the extracted features and the specific patterns of these categories, we were able to separate them during classification. The locations of *Beaches* and *Deltas* are marked in blue, and green, respectively.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, as an extension of [1], we analyzed several protected areas all over Europe by a high- and a medium-resolution space-borne instrument (delivering SAR and multi-spectral images). The accuracy of the results was computed for each sensor by comparing the retrieved results with reference data and this accuracy is on average about 97%.

By exploiting the specific imaging details and the retrievable semantic categories of these three image types (TerraSAR-X, Sentinel-1, and Sentinel-2), we can semantically fuse the image classification maps. In order to verify the classification results, we had to compare them with in-situ data.

For future evaluation, we plan to compare the classification accuracy of the wind turbines considering more parameters such as: the size of the pylons, the blade angles of the wind turbines, the rotation rates of the propellers, and the viewing direction and the resolution of the satellite images.

At the moment, there exist some studies about wind turbines [44][45][46] using SAR images but none of the existing papers analyzes all these parameters simultaneously. Therefore, we plan to compare the results from the point of view of accuracy between high-resolution vs. medium-resolution and between SAR vs. multi-spectral sensors.

In future, we plan to compare the performances acquired so far with the ones of the new system that will be developed under the CANDELA project [47].

In addition, depending on user feedback and responses by interested institutions, we could offer our software package to national and international authorities to support their coastline monitoring and disaster handling activities.

ACKNOWLEDGEMENTS

This work was supported by the H2020 ECOPOTENTIAL project. We thank the TerraSAR-X Science Service System for the provision of images (Proposals MTH-1118 and LAN-3156).

REFERENCES

- [1] C.O. Dumitru, G. Schwarz, and M. Datcu, “Monitoring of Coastal Environments Using Data Mining”, in Proc. of the International Workshop on Knowledge Extraction and Semantic Annotation (KESA), Athens, Greece, pp. 1-6, 2018.
- [2] T. Stepinski, P. Netzel, and J. Jasiewicz, “LandEx-A GeoWeb Tool for Query and Retrieval of Spatial Patterns in Land Cover Datasets”, IEEE JSTARS, 7(1), pp. 257-266, 2014.
- [3] C.R. Shyu, M. Klaric, G. Scott, A. Barb, C. Davis, and K. Palaniappan, “GeoIRIS: Geospatial Information Retrieval and Indexing System – Content Mining, Semantics Modelling, and Complex Queries”, IEEE TGRS, 45(4), pp. 839-852, 2007.

- [4] N. Boujemaa, "Ikona: Interactive Specific and Generic Image Retrieval", in Proc. of MMCBIR, Glasgow, UK, pp. 1-4, 2001.
- [5] M. Datcu et al., "Information Mining in Remote Sensing Image Archives: System Concepts", IEEE TGRS, 41(12), pp. 2923-2936, 2003.
- [6] TELEIOS project [accessed November 2018]. Available: <http://www.earthobservatory.eu/>.
- [7] EOLib project, [accessed November 2018]. Available: <http://wiki.services.eoportal.org/tiki-index.php?page=EOLib>.
- [8] J. Zhang, W. Hsu, and M.L. Lee, "Image Mining: Trends and Developments", 2002. [accessed June 2018]. Available: <http://www.comp.nus.edu.sg/~whsu/publication/2002/JIIS.pdf>.
- [9] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data Mining with Big Data", IEEE TKDE, 26(1), pp. 97-107, 2014.
- [10] ECOPOTENTIAL project, 2017. [accessed November 2018]. Available: <http://www.ecopotential-project.eu/>.
- [11] ECOSTRESS (Ecological Coastal Strategies and Tools for Resilient European Societies) project, [accessed March 2018]. Available: <http://ecostress.eu/pilot-areas/dutch-german-wadden-sea/>.
- [12] Wadden Sea World Heritage, 2017. [accessed March 2018]. Available: <http://www.waddensea-worldheritage.org/>.
- [13] K.S. Dijkema, J.H. Bossinade, P. Bouwsema, and R.J. de Glopper, "Salt Marshes in the Netherlands Wadden Sea: Rising High-Tide Levels and Accretion Enhancement", in "Expected Effects of Climatic Change on Marine Coastal Ecosystems", Kluwer Publishers, Dordrecht; pp 173-188, 1990.
- [14] S. Brusch and S. Lehner, "Monitoring River Estuaries and Coastal Areas using TerraSAR-X", in Proc. of OCEANS, Bremen, Germany, pp. 1-4, 2009.
- [15] S. Wiehle and S. Lehner, "Automated Waterline Detection in the Wadden Sea Using High-Resolution TerraSAR-X Images", Hindawi Journal of Sensors, pp. 1-6, 2015.
- [16] G. Heygster, J. Dannenberg, and J. Notholt, "Topographic Mapping of the German Tidal Flats Analyzing SAR Images with the Waterline Method", IEEE TGRS, 48(3), pp. 1019-1030, 2010.
- [17] M. Gade and S. Mechionna, "The Use of High-Resolution RADARSAT-2 and TerraSAR-X Imagery to Monitor Dry-Fallen Intertidal Flats", in Proc. of IGARSS, Quebec, Canada, pp. 1218-1221, 2014.
- [18] Danube Delta, 2016. [accessed March 2018]. Available: <http://romaniatourism.com/danube-delta.html>.
- [19] Danube Delta World Heritage, 2017. [accessed March 2018]. Available: <http://whc.unesco.org/en/list/588>.
- [20] S. Niculescu, C. Lardeux, I. Grigoras, J. Hanganu, and L. David, "Synergy between LiDAR, RADARSAT-2, and Spot-5 Images for the Detection and Mapping of Wetland Vegetation in the Danube Delta", IEEE JSTARS, 9(8), pp. 3651-3666, 2016.
- [21] M. Mierla, G. Romanescu, I. Nichersu, and I. Grigoras, "Hydrological Risk Map for the Danube Delta – A Case Study of Floods Within the Fluvial Delta", IEEE JSTARS, 8(1), pp. 98-104, 2015.
- [22] R. Tanase, A. Radoi, M. Datcu, and D. Raducanu, "Polarimetric SAR Data Feature Selection using Measures of Mutual Information," in Proc. of IGARSS, Milan, Italy, pp. 1140-1143, 2015.
- [23] P. Gastescu, "The Danube Delta Biosphere Reserve. Geography, Biodiversity, Protection, Management", Romanian Journal of Geography, 53(2), pp. 139-152, 2009.
- [24] D. Vaičiūtė, I. Olenina, R. Kavolytė, I. Dailidienė, and R. Pilkaitytė, "Validation of MERIS chlorophyll a products in the Lithuanian Baltic Sea case 2 coastal waters", in Proc. of IEEE/OES Baltic International Symposium (BALTIC), Riga, Latvia, pp. 1-2, 2010.
- [25] G. Garnaga and Z. Stukova, "Contamination of the south-eastern Baltic Sea and the Curonian Lagoon with oil products," in Proc. of IEEE/OES US/EU-Baltic International Symposium, Tallinn, Estonia, pp. 1-8, 2008.
- [26] S. Gulbinskas, E. Trimonis, and I. Mineviciute, "Sedimentary fluxes in the marine-lagoon (Baltic sea – Curonian Lagoon) connection," in Proc. of IEEE/OES Baltic International Symposium (BALTIC), Riga, Latvia, pp. 1-6, 2010.
- [27] J. Maytas, "Using SAR Data for Wet Snow Monitoring", Diploma thesis, Charles University, Prague, 2013. [accessed June 2018]. Available: https://dspace.cuni.cz/bitstream/handle/20.500.11956/71298/DPTX_2_013_2_11310_0_363946_0_151281.pdf?sequence=1.
- [28] P. Trujillo, C. Piroddi, J. Jacquet, "Fish Farms at Sea: The Ground Truth from Google Earth", 2012. [accessed March 2018]. Available: <https://doi.org/10.1371/journal.pone.0030546>.
- [29] Mediterranean basin: Fish farms at Sea, 2017. [accessed March 2018]. Available: <http://www.fao.org/fishery/naso-maps/selected-aquaculture-sites/mediterranean-basin/en/>.
- [30] Sentinel-1 counts fish, 2018. [accessed June 2018]. Available: https://www.esa.int/Our_Activities/Observing_the_Earth/Sentinel-1_counts_fish.
- [31] J. D. Ballester-Berman, P. Sanchez-Jerez, and A. Marino, "Detection of aquaculture structures using Sentinel-1 data", in Proc. of EUSAR, Aachen, Germany, pp. 1-4, 2018.
- [32] FAO of the United Nations, 2018. [accessed June 2018]. Available: <http://www.fao.org/fishery/en>.
- [33] C.O. Dumitru, G. Schwarz, and M. Datcu, "Land Cover Semantic Annotation Derived from High Resolution SAR Images", IEEE JSTARS, 9(6), pp. 2215-2232, 2016.
- [34] C. Dumitru, G. Schwarz, and M. Datcu, "SAR Image Land Cover Datasets for Classification Benchmarking of Temporal Changes", IEEE JSTARS, 11(5), pp. 1-21, 2018.
- [35] In-situ data, wind turbines, 2018. [accessed June 2018]. Available: <https://www.flickr.com/photos/94515068@N04/8603945668/>.
- [36] In-situ data, boats/cargo ship, 2018. [accessed June 2018]. Available: <https://www.flickr.com/photos/meshal/13200620583/>.
- [37] In-situ data: fish cages, 2018. [accessed June 2018]. Available: <https://flic.kr/p/aDYS8v>.
- [38] In-situ data, aquaculture, 2018. [accessed June 2018]. Available: <https://www.flickr.com/photos/johnbostock/6144644349/>.
- [39] In-situ data: dams, 2018. [accessed June 2018]. Available: <https://www.natuurmonumenten.nl/projecten/marker-wadden/english-version>.
- [40] In-situ data: dunes, 2018. Photo by Alma de Groot, "Aeolian sand transport on Romo". [accessed June 2018]. Available: <http://qsr.waddensea-worldheritage.org/reports/beaches-and-dunes>.
- [41] In-situ data: tidal flats, 2018. [accessed June 2018]. Available: <https://flic.kr/p/VZ5mzA>.
- [42] In-situ data: beaches, 2018. [accessed June 2018]. Available: <https://www.flickr.com/photos/iahimr/6171124767/>.
- [43] In-situ data, deltas, 2018. [accessed June 2018]. Available: <https://www.flickr.com/photos/cost3l/15982917459/>.
- [44] C. Clemente and J.J. Soraghan, "Analysis of the effect of wind turbines in SAR images," in Proc. of IET International Conference on Radar Systems (Radar 2012), Glasgow, UK, pp. 1-4, 2012.
- [45] T. Cuong, "Radar cross section (RCS) simulation for wind turbines", Master Thesis, Naval Postgraduate School, California, 94 pages, 2013.
- [46] M.B. Christiansen and Ch.B. Hasager, "Wake effects of large offshore wind farms identified from satellite SAR", Remote Sensing of Environment, 98(2-3), pp. 251-268, 2005.
- [47] CANDELA project [accessed November 2018]. Available: <http://www.candela-h2020.eu/>.

Automated Continuous Data Quality Measurement with QuaIle

Lisa Ehrlinger^{*†}

[†]Software Competence Center Hagenberg GmbH
Softwarepark 21, 4232 Hagenberg, Austria
email: lisa.ehrlinger@scch.at

Bernhard Werth^{*}, Wolfram Wöß^{*}

^{*}Johannes Kepler University Linz
Altenberger Straße 69, 4040 Linz, Austria
email: lisa.ehrlinger@jku.at, bernhard.werth@fh-hagenberg.at,
wolfram.woess@jku.at

Abstract—Data quality measurement is essential to gain knowledge about data used for decision-making and to evaluate the trustworthiness of those decisions. Example applications, which are based on automated decision-making, are self-driving cars, smart factories, and weather forecast. One-time data quality measurement is an important starting point for any data quality project to detect critical data that does not meet expectations and to define improvement goals for data cleansing activities. The complementary task of *continuous* data quality measurement is essential to ensure that data continues to conform to requirements and to detect unexpected changes in the data. However, most existing data quality tools allow quality measurement at a specific point in time while leaving the automation and scheduling to the user. In this paper, we highlight the need for (1) domain-independent ad hoc measurement, to provide a quick insight of an information system’s qualitative condition, and (2) continuous data quality measurement, to observe how data quality evolves over time. Both requirements can be achieved with our data quality tool QuaIle (Quality Assessment for Integrated Information Environments, pronounced [ˈkvalə]), which we developed to calculate metrics for the quality dimensions accuracy, correctness, completeness, pertinence, timeliness, minimality, readability, and normalization on both data-level and schema-level. The quality measurements can be either exported as a user- and machine-readable quality report, or they can be periodically stored in a database, which allows for long-term analysis. In this paper, we demonstrate the application of QuaIle for ad hoc and continuous data quality measurement.

Index Terms—Data Quality; Measurement; Monitoring; Estimation; Trust.

I. INTRODUCTION

Strategic decisions are usually based on data. Examples are human-made decisions in enterprises whether to promote or to suspend the production of a specific product due to sales data. In the era of artificial intelligence, machine-based decisions gain increasing relevance in applications like self-driving cars, search engines, or industry robots. In order to trust such data-driven decisions, it is necessary to measure and know the quality of the underlying data with appropriate tools [1]. Despite the clear correlation between data and decision quality, 84 % of the CEOs in the US are concerned about their data quality [2]. In addition to incorrect decision making, poor Data Quality (DQ) may cause effects like cost increase, customer dissatisfaction, and organizational mistrust [3]. According to an estimation by IBM, the total financial impact of poor quality data on business in the US was \$3.1 trillion [4] in 2016. Thus, DQ is no longer a question of “hygiene”, but has become critical for operational excellence and is perceived as the greatest challenge in corporate data management [5].

We originally presented the Java-based DQ tool QuaIle in [1], which automatically performs domain-independent quality measurement on both data-level and schema-level. The name “QuaIle” is not only an abbreviation for “Quality Assessment for Integrated Information Environments”, but it is also the German word for jellyfish. We consider this amazing but yet not fully explored animal a proper representative for our DQ tool, since its complex life cycle consists of two major stages, which are divided by two different reproduction phases: (1) the stationary polyp-phase, and (2) the free-swimming medusa-phase [6]. In analogy to the jellyfish life cycle, the two major aims of QuaIle are:

- (1) Initial ad hoc data quality measurement (*stationary*)
- (2) Continuous data quality measurement (*free-swimming*)

Both phases are necessary to provide holistic DQ measurement. In the stationary polyp-phase, a machine- and human-readable XML (extensible markup language) quality report is generated, which allows to grasp a first insight into the qualitative condition of an information system (IS) without requiring preparation activities or deeper domain-knowledge. Initial ad hoc DQ measurement guides the path for more detailed investigation and target-oriented data profiling. The medusa-phase allows the long-term observation of an IS’s quality in order to detect trends or outliers over time. Continuous DQ measurement provides indications for further DQ improvements and thus, increases the trustworthiness for data-driven decisions. While the polyp-phase was originally introduced in [1], the medusa-phase is the major contribution of this paper. In addition, we present several implementation extensions of QuaIle: a new data source connector for Apache Cassandra [7], new metrics for the DQ dimensions timeliness and readability, and experiments with large and highly volatile real-world data.

Jellyfish also contribute to science with their green fluorescent protein (GFP), which can be used as marker for gene expressions or as reporter for virus infections [8]. QuaIle marks DQ issues for the quality dimensions accuracy, correctness, completeness, pertinence, timeliness, minimality, readability, and normalization. Since data of enterprises and organizations are usually stored in Integrated Information Systems (IISs) [9], an important feature of a DQ tool is to estimate the quality of different and often heterogeneous ISs on-the-fly to select the most appropriate and most trustworthy source for a given query. The jellyfish tentacles of QuaIle allow virtual integration of different data sources or parts of data sources [1].

While a number of DQ tools has been proposed over the years (cf. [10][11][12][13]), most domain-independent tools focus on data cleansing and/or data integration. We want to highlight that our focus is the quality measurement of an IIS in productive use, and automatic data cleansing activities are not in the scope of this research work. In order to understand the degree and effectiveness of data cleansing and to define goals for further cleansing activities, it is necessary to measure and know the quality of the data [14]. To the best of our knowledge, there exists no tool that offers (continuous and ad hoc) DQ measurement for such a large number of different DQ dimensions in a single application and comprises both data and schema quality. Thus, the main contributions of QuaIle in contrast to other DQ tools can be summarized as follows:

- Domain-independent ad hoc DQ measurement
- Continuous data quality measurement
- Virtual data integration as basis for DQ estimation
- Combination of data and schema quality measurement

The remainder of this paper is organized as follows: in Section II, we discuss the state-of-the-art into data quality, clarify the concept of “continuous data quality measurement” and differentiate QuaIle from existing DQ tools. Section III covers all data and schema quality dimensions and metrics, which were applied in this research. In Section IV, we describe the implementation of QuaIle and demonstrate its application in terms of ad hoc and continuous DQ measurement.

II. DATA QUALITY: STATE OF THE ART

Data quality is usually defined by the “fitness for use” principle [15][16][17], which refers to the high subjectivity and context-dependency of this topic. This definition emphasizes the fact that DQ cannot be evaluated without considering the context (e.g., a specific application or service), in which the data is used for. To grasp single qualitative aspects on the data, DQ is typically described by a set of DQ *dimensions* (e.g., accuracy, completeness, timeliness) and *metrics*, which are specific formulas to calculate the dimensions [15][18]. A methodology for DQ can be divided into the following activities [19][20]: (1) state reconstruction, (2) DQ measurement, (3) data cleansing or improvement, and (4) continuous DQ measurement or monitoring. Step (1) comprises the collection of contextual information on the observed data and the organization where a DQ project is carried out [19]. DQ measurement (2) is typically described by a set of DQ dimensions and assigned metrics. Until now, there exists no agreement on a standardized set of dimensions and metrics for DQ measurement [14]. Data cleansing (3) is the process of correcting erroneous data and includes tasks like customer data standardization or data de-duplication. According to [20], automated data cleansing methods are very valuable for Big Data, but with the risk to insert new errors. Step (4) is mainly used implicitly in literature in order to refer to the ongoing observation of an IS’s quality and is discussed in more detail in Section II-A. While we use the term “continuous DQ measurement” to describe this DQ activity, synonymously

used terms are “recurrent DQ assessment” [20], “ongoing DQ measurement” [14], or “automated DQ monitoring” [21]. The importance for *automation* in data quality projects is highlighted in [14], for both, initial and ongoing DQ measurement. Without automation, the volume and volatility of data in complex real-world IS will quickly overwhelm any DQ measurement efforts [14].

A. Continuous Data Quality Measurement

The term “continuous data quality measurement” (CDQM) describes the calculation and storage of DQ metrics over time, in order to ensure that the qualitative condition of the data remains stable [22]. Sebastian-Coleman [14] distinguishes between in-line measurement, periodic measurement, and controls. In-line and periodic measurement are distinguished by the measurement frequency, where in-line measurement is applied to critical or highly volatile data, and periodic measurement is used to monitor less frequently updated data (e.g., master data) [14]. Our implementation QuaIle allows to apply both types: in-line as well as periodic DQ measurement. Table I describes the most important requirements for CDQM defined in [22] and how they are implemented in QuaIle.

A control is understood as a built-in data check, which is typically implemented in a data transformation (or integration) process [14]. We want to point out that the majority of DQ tools that state to offer *monitoring* functionality refer to controls. Also Gartner [10] describe the term “monitoring” as “the deployment of controls in order to ensure that data continues to conform to business rules”. We explicitly want to distinguish our understanding of CDQM (i.e., in-line and periodic measurement), as it is implemented in QuaIle, from DQ monitoring using controls. A practical example of a monitoring tool with controls in the physics domain is the DQ monitoring framework for the ATLAS experiment at the Large Hadron Collider at CERN [27]. Before the automatically collected data is shipped to the data store, a number of pre-defined quality checks are carried out in order to ensure that the data is free of error and can be used for scientific data analysis. This DQ tool is accompanied by human domain experts, who inspect the generated visualization and take action in case of problems.

In accordance with [14], we want to highlight that the usefulness and application of CDQM depends on the volatility of the data set. We distinguish between three volatility types:

- (V1) *Static data sets*, which are very rarely or not modified, for example, a list of planets in the solar system or countries and their capitals.
- (V2) *Periodically or occasionally updated data*, for example, master data like products, customers, employees, or the daily menu of a restaurant.
- (V3) *Highly volatile data*, for example, stock exchange prices, sales data from large vendors, streaming or sensor data.

All three types require different measurement strategies, which are demonstrated with QuaIle in Section IV. The

TABLE I
REQUIREMENTS FOR CONTINUOUS DATA QUALITY MEASUREMENT

Requirement	Description	Implementation in QuaIle
DQ measurement functionality	To define how DQ should be actually measured is not trivial and according to [14], one of the biggest challenges for DQ practitioners. One reason is the ongoing discussion on DQ dimensions, where until now, no agreement could be reached [14].	QuaIle currently implements metrics for eight different DQ dimensions on both data-level and schema-level. Since not all published DQ metrics have been sufficiently evaluated so far (cf. [23]), QuaIle allows to evaluate metrics, e.g., whether they are suited for CDQM or not.
Storage	DQ measurements must be persisted to allow DQ analysis over time and to compare current DQ measurements to older ones.	For the experiments in this paper, we used a MySQL database to store CDQM results, but any kind of database might be used.
Automation	DQ measurement should be performed automatically, based on user-defined time periods.	We automated DQ measurements either directly in a Java method, or with Windows Task Scheduler [24].
Analysis	Visual as well as statistical (time-series) analysis is required to present the DQ measurement results and to detect patterns and changes in DQ measurements.	Since a graphical user interface for QuaIle is currently under development, we performed the analysis of CDQM with the Python packages pandas [25] and matplotlib [26].

strategies differ in the measurement frequency (in-line versus periodic), in the amount of data to be included in the measurement, as well as in the automatically triggered action in case a specific threshold is not met.

B. Data Quality Tools

Despite a continuous growth, the market of DQ tools is still considered a niche market [10]. Gartner lists 39 commercial DQ tools by 16 vendors in their “Magic Quadrant of Data Quality Tools 2017” [10]. Most of the tools offer functionalities to investigate the qualitative condition of different data sources using data profiling techniques, manage DQ rules, resolve DQ issues, enrich data quality by integrate external data, validate addresses, standardize and cleanse data, and link related data entries using a variety of techniques. The aim of these commercial tools is usually the support of a comprehensive DQ program that involves management, IT, and business users. Thus, the application of such a tool usually requires a domain expert and preparatory work to be effective.

In addition to commercial DQ tools, a number of scientific tools have been proposed over the years, where the most important ones are compared and discussed in [11][12][13]. All three surveys make clear that the focus of those tools is on the detection and cleansing of specific DQ problems (e.g., name conflicts, missing data) and none of the observed tools provide any monitoring functionality. Examples for typical data cleansing tools are Potter’s Wheel [28] or Wrangler [29]. In contrast to the tools observed in existing surveys [11][12][13], QuaIle focuses on the pure measurement (detection) of DQ problems and does not cleanse data. The advantage of DQ measurement without cleansing or manipulating data is the potential for domain-independent automation, that is, it can be performed unsupervised and ad hoc without any consequences to insert new errors in the data.

Additionally, and in contrast to most existing DQ tools (except for [30] and [31], which will be discussed in the following paragraph) QuaIle addresses the DQ topic from the dimension-oriented view. While a lot of research on DQ dimensions and their definitions has been proposed in literature [3][15][17], there is no tool that implements generally-applicable metrics

for such a broad number of dimensions. QuaIle fills this gap and can thus be considered a vital complement in the section of research-oriented DQ tools. Of course, more specialized tools might outperform QuaIle in specific implementations, like distance calculation or string matching.

In terms of CDQM capabilities and the DQ dimension-oriented view, we found two open source tools that can be compared to QuaIle: MobyDQ by Alexis Rolland [30] (formerly: “Data Quality Framework”) and Apache Griffin [31], a project from the Apache Incubator. However, both tools require an intensive configuration phase, where DQ metrics and checks are defined depending on the observed domain and data. No metrics for initial ad hoc measurement are provided. While MobyDQ is easy to install, Apache Griffin is very arduous to set up, because it depends on several other open source tools that are still in the incubator status. We needed six days for the installation until we were able to produce usable DQ measurements.

III. DATA AND SCHEMA QUALITY DIMENSIONS

Data quality is usually described as multidimensional concept, which is characterized by different aspects, so called *dimensions* [15]. Those dimensions can either refer to the data values (i.e., *extension* of the data), or to their schema (i.e., the *intension* or data structure) [18]. While the majority of research into DQ focuses on the data values, QuaIle implements DQ measurements for both schema and data values. In fact, schema quality has a strong impact on the quality of the data values [18]. An example are redundant schema elements, which can lead to data inconsistencies. Thus, it is essential to consider both topics in order to provide holistic DQ measurement.

Since a wide variety of quality dimensions has been proposed over the years (e.g., [15][17][32][33]), we focus in the following paragraphs on the eight dimensions (1) accuracy, (2) correctness, (3) completeness, (4) pertinence, (5) timeliness, (6) minimality, (7) readability, and (8) normalization. Each dimension can be quantified with one or several metrics, which capture the fulfillment of a dimension in a numerical value [34]. Heinrich et al. [23] defined five requirements a

data quality metric must fulfill. The implemented metrics in QuaIle can thus be evaluated by means of the requirements in [23] or if they are suited for application in CDQM.

Some metrics require a reference or benchmark (*gold standard*) for their calculation. According to the Oxford Dictionary, a Gold Standard (GS) is “the best, most reliable, or most prestigious thing of its type” [35]. In the vast majority of cases a gold standard does not exist, but if there is one, it would be used in place of the IS under investigation. Thus, in practice, an existing benchmark is employed as gold standard, e.g., a single IS can be compared to the integrated data from the complete IIS. Although in practice, there is usually no complete gold standard for large data sets available, there are often reference data sets of good quality for a subset of the data. Examples are purchased reference data sets for customer addresses or a manually cleaned part of the original data. The quality estimation in QuaIle (cf. Section III-H) allows to extrapolate the exact measurement for a part of the data to other parts that are required for a query but have not been yet measured. For more details to the schema quality dimensions applied in this paper, we refer to [36] and more information on the data quality dimensions can be found in [37].

A. Accuracy and Correctness

The terms *accuracy* and *correctness* are often used synonymously in literature and a number of different definitions exist for both terms [15][18][38]. In the DQ literature, accuracy can be described as the closeness between an information system and the part of the real-world it is supposed to model [18]. From the natural sciences perspective, accuracy is usually defined as the magnitude of an error [38]. In this research work, we refer to correctness for a calculation, which has been presented by Logan et al. [39], who distinguish between correct (C), incorrect (I), extra (E) and missing (M) elements after comparing a data set to its reference:

$$Cor(c, c') = \frac{C}{C + I + E}. \quad (1)$$

Here, the data correctness of, for instance, a relational table or class in an ontology, denoted as concept c , is measured by comparing it to its “correct” version c' . In this notion, C is the number of elements that correspond exactly to an element from the reference c' . The incorrect elements I have a similar element in the gold standard, but are not identical. While M describes the number of missing elements in the IS under investigation that exist in the gold standard, its complement E is the number of extra elements that exist in the investigated IS, but have no corresponding element in the gold standard. We show below that metrics for correctness and pertinence can be determined by the same basic element counts (C, I, E, M – short CIEM), which allows an efficient implementation. Algorithm 1 demonstrates how the element counts are calculated.

On the data-level, QuaIle implements an accuracy metric, which has its origins in the field of machine learning and

Algorithm 1: Calculation of CIEM Counts

Input: Data set ds , its reference ds' , and threshold t .

Output: The number of correct C , incorrect I , extra E , and missing M elements.

```

1 for each element  $e_1$  in  $ds$  do
2   for each element  $e_2$  in  $ds'$  do
3      $\sigma = \text{calculateSimilarity}(e_1, e_2)$ ;
4     if  $\sigma == 1.0$  then
5       setUpCorrectAssignment( $e_1, e_2$ );  $C++$ ;
6     else if  $\sigma > t$  then
7       setUpIncorrectAssignment( $e_1, e_2$ );  $I++$ ;
8     else
9        $E++$ ;
10    end
11  end
12 end
13 for each element  $e_2$  in  $ds'$  do
14   if  $e_2$  has no assignment then
15      $M++$ ;
16   end
17 end
```

is usually used to measure the accuracy of classification algorithms [40]. This accuracy metric can also be mapped to the notion by Logan et al. [39]:

$$Acc(c, c') = \frac{|c|}{|c \cup c'|} = \frac{C}{C + I + E + M}, \quad (2)$$

where $|c|$ gives the number of records in a data set or concept c . In the rest of this paper, we refer to accuracy when discussing quality metrics for data values (since QuaIle implements the metric for accuracy on data-level), and to correctness when discussing the corresponding schema dimension.

On the schema-level, Vossen [41] describes a database (DB) schema as correct, if the concepts of the related data model are applied in a syntactically and semantically appropriate way, effectively considering only the model as reference. In [18] the authors distinguish between correctness with respect to the model and with respect to requirements. Although the values of an IS might also be erroneous, the content of an IS can be added as third possibility to validate a schema. Three types of validation are therefore possible:

- Validation of a schema against its conceptual *model* (e.g., Entity-Relationship model) assumes the correct representation of the modeled constructs within the schema.
- When a schema is validated against its *requirements*, the requirements are expected to be represented correctly. This is usually considered to be a manual task (cf. [42]), because requirements are rarely available in machine-readable form.
- Validation against the *content* of an information source verifies whether the schema fits its values. This includes for instance the correct usage of attributes (e.g., an

attribute `first_name` actually contains a person's first name and no numeric value).

In QuaIle, the formula by Logan et al. [39] for data correctness is also employed as a metric for schema correctness with C_s , I_s , E_s , and M_s denoting the correct, incorrect, extra, and missing elements of a schema s :

$$Cor(s, s') = \frac{C_s}{C_s + I_s + E_s}. \quad (3)$$

B. Completeness

Completeness is broadly defined as the breadth, depth, and scope of information contained in the data [17] and can be divided into three subtypes. *Schema completeness* is the degree to which concepts and their attributes are present in a schema, *column completeness* defines the ratio of missing values to all values in a variable, and *population completeness* measures the missing values with respect to the real reference population [18]. A number of authors [15][18] calculate data completeness according to:

$$Com(c, c') = \frac{|c|}{|c'|}. \quad (4)$$

Despite differences in expressions, most existing completeness metrics are correspondent to (4) and compare the number of elements in a data set $|c|$ to the number of elements in the gold standard $|c'|$. In this metric, scope for interpretation lies in selecting the gold standard or reference c' and in the similarity calculation (i.e., determining whether an element has a reference element in c'). In QuaIle however, extra records, which exist in the gold standard, but have no counterpart in the data set under investigation are excluded and therefore have no influence on the completeness calculation. We use the formula presented by Logan et al. [39]:

$$Com(c, c') = \frac{C + I}{C + I + M}. \quad (5)$$

In addition, QuaIle provides a variant of (4) that explicitly ignores duplicate entries:

$$Com(c, c') = \frac{|\text{unique}(c)|}{|\text{unique}(c')|}. \quad (6)$$

Oliveira et al. [43] provide the only formal definition of completeness (i.e., the missing value problem) for a relational schema $R(A)$ that contains a finite set of relations $r(A)$, where A is an attribute set (a_1, a_2, \dots, a_m) , t a tuple, and $v(t, a)$ a specific value for tuple t and attribute a :

Definition 1: Let S be a set of attribute names, defined as: $S = \{a | a \in R(A) \wedge a \text{ is a mandatory attribute}\}$, i.e., $S \subseteq R(A)$. There is a missing value in attribute $a \in S$ iff: $\exists t \in r : v(t, a) = \text{null}$.

In contrast to other approaches, Definition 1 is restricted to null values and does not consider default or placeholder values (e.g., "NA" or "-99") as data incompleteness. Hinrichs

proposed a metric, which corresponds to Def. 1 for different aggregation levels: attribute-value-level, record-level, concept-level, and DB-level. While the completeness of an attribute value $Com(v)$ is either 0 (if $v=\text{null}$) or 1 (else), completeness on record-level $Com(r)$ is the arithmetic mean of all attribute-value completeness measures for that record. The completeness of a concept (relation in [44]) is defined as

$$Com(c) = \frac{\sum_{i=1}^n Com(r_i)}{|c|}, \quad (7)$$

where n is the number of records in concept c . In addition, DB-level completeness is defined as the arithmetic mean of all concept-level completeness measures.

Schema completeness describes the extent to which real-world concepts of the application domain and their attributes and relationships are represented in the schema [18]. The metric for schema completeness in QuaIle corresponds to the metric for data completeness in (5):

$$Com(s, s') = \frac{C_s + I_s}{C_s + I_s + M_s}. \quad (8)$$

Batista and Salgado [45] applied a schema completeness metric, which is equivalent to the data completeness in (4). In the calculation, the number of elements in the reference schema $|s'|$ is determined by counting the number of distinct elements in all schemas of an IIS. While the authors in [45] assume pre-defined schema mappings to be provided, QuaIle calculates the distance or similarity values between the schema elements on-the-fly.

In addition, Nauman et al. [46] proposed a comprehensive IIS completeness metric, which incorporates the *coverage* (i.e., data completeness of the extension of an IS), and *density* (i.e., schema completeness of the intension of an IS). The authors use the entire IIS as gold standard. The density of a schema is calculated according to the population of attributes with non-null values [46]. In contrast, the schema completeness metric in QuaIle implements a data-value-independent calculation, which considers the existence of specific schema elements (e.g., relations in a relational DB).

C. Pertinence

Pertinence on the data-level equates to the notion of precision (in contrast to recall [40]) from the information retrieval field and complements data completeness. Data pertinence describes the prevalence of unnecessary records in the data. The classic precision metric is defined as the probability to select a correct element from a list [40] and in terms of correct, incorrect, extra, and missing records, is defined as:

$$Per(c, c') = \frac{C + I}{C + I + E}. \quad (9)$$

Schema pertinence describes a schema's relevance, which means that a schema with low pertinence has a high number of unnecessary elements [18]. A schema that is perfectly

complete and pertinent represents exactly the reference schema (i.e., its real world representation), which means that the two dimensions complement each other. In accordance to (9), schema pertinence is calculated in QuaIle as

$$Per(s, s') = \frac{C_s + I_s}{C_s + I_s + E_s}, \quad (10)$$

where the number of schema elements with a (correct or incorrect) correspondence in the gold standard is divided by the total number of elements in the schema under investigation.

D. Timeliness

An important aspect of many types of data is that the data values may change over time, for example, product pricing or customer addresses. In terms of time-related DQ dimensions, it can be distinguished between *currency* and *timeliness* [18]. According to [18], currency describes how promptly data is updated compared to changes in the real world. Ballou et al. [47] proposed the following metric for the currency of a single record r :

$$Cur(r) = (DeliveryTime(r) - InputTime(r)) + Age(r), \quad (11)$$

which requires meta information about each record. *DeliveryTime* is the time when the data is delivered to the customer, *InputTime* describes the time when the data is entered in the IS, and *Age* is the age of the data prior to system entrance. Since the age is rarely available and the delivery time would require in-depth domain knowledge, currency is calculated in QuaIle as

$$Cur(c) = \frac{\sum_{r \in c} Now - InputTime(r)}{|c|}. \quad (12)$$

Here, *Age* is assumed to be 0 and the delivery time is assumed to be the timestamp of the data quality assessment. Additionally, the record-wise currency values are aggregated via average to assess the currency of a data set (i.e., concept). According to [18], timeliness describes how current the data is for a task at hand, which takes into account the specific use of a data value. An example would be the program of a cinema, which could be current because it contains the most recent data, but it is not timely if it is available after the start of the desired movie. Since QuaIle focuses on the objectively measurable quality dimensions, timeliness is calculated according to:

$$Tim(c) = \max\left(0, 1 - \frac{Cur(c)}{Vol(c)}\right), \quad (13)$$

omitting the aspect of the data usage. *Vol(c)* is the *volatility* of a data set, which is a domain-specific value that describes how fast records become irrelevant [47]. Stock exchange prices (V3) that are updated by the second are considered highly volatile, while customer addresses display a low volatility as

customers move every few years at maximum. Some types of data like birth dates have a volatility of 0, because they never become obsolete, therefore infinite timeliness is assumed in QuaIle.

While schema evolution might cause a schema to become outdated, to the best of our knowledge, no approaches for measuring the outdatedness of schemas exist. Therefore, the current version of QuaIle considers schemas to be time-invariant. It should be pointed out that this would be an interesting topic to be considered as future work.

E. Minimality

Information sources are considered minimal if no parts of them can be omitted without losing information, that is, the IS is without redundancies and no duplicate records exist [18]. The detection of duplicate records is a widely researched field that is also referred to as record linkage, data deduplication, data merging, or redundancy detection [48]. In order to determine which records of a data set are duplicates, different approaches exist. The most prominent approaches can be assigned to one of the following types [48]: (1) *probabilistic assignment* using the Fellegi-Sunter model [49], (2) *machine learning techniques* like support vector machines, clustering algorithms, or decision trees, (3) *distance-based methods*, which are based on a function that calculates the distance between two objects, and (4) *rule-based methods*, which are usually based on the work of domain experts.

In QuaIle, duplicate detection is done by hierarchical clustering, which requires a distance function between the records. A distance function $\delta : o \times o \rightarrow [0, 1]$ is a function from a pair of elements to a normalized real number expressing the distance or dissimilarity between the two elements [50]. Analogous, some techniques calculate the similarity $\sigma : o \times o \rightarrow [0, 1]$ between two elements, which can be transformed to a distance value using the formula $\delta = 1 - \sigma$. All distance and similarity values produced by QuaIle can be assumed to be normalized over the unit interval of real numbers [0,1].

Since each data record consists of multiple attribute values, the distance function is a weighted-average of individual attribute distance functions. QuaIle offers the following distance functions for data values: *AffineGapDistance*, *CosineDistance*, *LevenshteinDistance*, and *SubstringDistance* for strings, *AbsoluteValueDistance* for double values, *EqualRecordDistance* for entire records, as well as *EnsembleDistance* for any data type. The latter one combines an arbitrary number of other distances and adds a weight for each one. Thus, it allows the creation of distances that are adjusted to a specific IS schema, for example, to calculate the distance between persons by applying a string distance to the first and last name and a distance for numeric attributes to the age, and giving higher weights to the name than the age.

The main advantage of clustering in our approach is the automatic resolution of multiple correspondences. It thus,

however, requires a threshold to be defined. QuaIle sets a predefined clustering threshold, which has been evaluated in experiments presented in [36]. In an automated test run, similarity matrices with different parameter combinations have been compared to a similarity matrix created by a domain expert using the mean squared error (MSE). The parameter combination yielding the closest similarity results (having a MSE of 0.0102) were used as standard parameters. However, QuaIle also allows to overwrite those values by the user to adjust for specific domains. Hierarchical clustering initially creates one cluster for each observed record and continuously combines different clusters until all records are subsumed into one large cluster. QuaIle offers seven different linkage strategies (single linkage, complete linkage, median linkage, mean linkage, pair group method with arithmetic mean, centroid linkage, and Ward's method). We refer to [51] for further information on hierarchical clustering.

Following, the minimality metric in QuaIle is based on a three-step approach, which is used for the data values and the schema elements likewise. Consequently, we refer to the observed objects as "elements", using the more generic term for both, records, as well as schema elements.

- 1) *Element-wise distance calculation.* All elements are compared to each other, which yields a distance matrix.
- 2) *Clustering.* All elements are hierarchically clustered according to their distance values. In a perfectly minimal IS, the number of elements $|c|$ should be equal to the number of clusters $|clusters|$. If two or more elements are grouped together into one cluster, the minimality score drops to a value below 1.0.
- 3) *Minimality calculation.* Finally, the minimality can be calculated according to

$$Min(c) = \begin{cases} 1.0, & \text{if } |c| = 1 \\ \frac{|clusters|-1}{|c|-1}, & \text{else} \end{cases} \quad (14)$$

Schema minimality is of particular interest in the context of IIS, where redundant representations are common. The minimality of a schema is an important indicator to avoid redundancies, anomalies and inconsistencies. QuaIle calculates schema minimality according to the three-step approach described above. For the schema similarity, the following distance functions are available: `DSDAttributeDistance` on attribute-level, `DSDConceptAssocDistance` on concept- or association-level, and `SimilarityFloodingDistance` on schema-level. DSD (data source description) is a vocabulary to semantically describe IS schemas [21] and is explained in more detail in Section IV-B. The first two distances are ensemble distances, which are adjusted to the DSD representation of attributes or concepts and associations respectively. In addition, we implemented the Similarity Flooding (SF) algorithm proposed in [52], which calculates the similarity between nodes in a graph-based schema representation, and can thus only be applied to a complete DSD schema (in contrast to

single concepts). Subsequently, (14) can be reformulated for schema minimality according to

$$Min(s) = \begin{cases} 1.0, & \text{if } |s| = 1 \\ \frac{|clusters|-1}{|s|-1}, & \text{else} \end{cases}, \quad (15)$$

where $|s|$ is the number of elements (concepts and associations) in a schema s .

F. Normalization

Normal Forms (NFs) can be used to measure the quality of relational DBs, with the aim of obtaining a schema that avoids redundancies and resulting inconsistencies as well as insert, update, and delete anomalies [41]. In contrast to all other schema quality dimensions listed in this paper, normalization requires access to the extension of the information source, i.e., the data values themselves. Although this quality dimension refers to relational data only, it is included in QuaIle, because of the wide spread use of relational DBs in enterprises. Several modern DBs use denormalization deliberately to increase read and write performance. Hence, depending on the type of IS, a NF evaluation is not always helpful in deducing the quality of its schema. It can however, serve as checking mechanism to ensure that only controlled denormalization exists.

Identifying *functional dependencies* (FDs) forms the basis for determining the NF of a relation. A FD $\alpha \rightarrow \beta$, where α and β are two attribute sets of a relation \mathcal{R} , describes that two tuples that have the same attribute values in α must also have the same attribute values in β . Thus, the α -values functionally determine the β -values [53].

In QuaIle, the second, third, and Boyce Codd normal form (2NF, 3NF, and BCNF, respectively) can be determined. The applied algorithm can be classified as a bottom-up method [54], in which the FDs of a relation are analyzed by comparing all attributes' tuple values with all other attributes' tuple values. Then, the minimal cover is determined by performing left- and right-reduction so that all FDs are in canonical form and without redundancies [41]. Following, all attributes are classified as key or non-key attributes and based on all information gathered, the correct NF is determined. Each schema element is annotated with quality information about its NF, key attributes, and minimal cover.

G. Readability

The current version of QuaIle supports readability on schema-level only. Although we did not find any discussion on readability on data-level, the current implementation of the readability could be used to evaluate the readability of string values on content-level likewise.

A schema should be readable, which means it should represent the modeled domain in a natural and clear way so it is self-explanatory to the user [41]. Good readability supports semantic expressiveness and enables automatic mappings to other domains that are based on dictionary approaches

(e.g., by using publicly available online dictionaries such as WordNet [55] or DBpedia [56]). While the readability of a conceptual schema in its graphical representation also includes aesthetic criteria, such as the arrangement of entities or crossing lines, the readability of a logical schema is limited to the actual naming of entities and relationships. Since clarity is subjective, no generally valid formal definition for this quality dimension exists [18].

We suggest two core measures to guarantee a sufficient level of readability. The first measure is based on a validation of all schema element names using a dictionary approach, where we selected WordNet [55] for the implementation in QuaIle. As a general rule, concepts should usually be described by singular nouns, while relationships should be described in present tense verbs or by a combination of two nouns [57]. A user should be permitted to add company-specific abbreviations that are widely used in practice and are usually not contained in public dictionaries. This is currently allowed in form of a CSV (comma-separated values) file. The second measure is a mandatory set of readability rules with which a rule checker can verify compliance. Both, readability rules and compliance with the online dictionary, can be formulated in criteria *crit*. Those criteria are applied to “words”, which are extracted from schema element labels (e.g., an attribute name). A word can be either the complete name of an element, or part of it. If a schema uses delimiters like underscores (_) or hyphens (-), one string is split into several words. For example, `first_name` is split into “first” and “name”. The following list contains a set of exemplary criteria, which can be used as a starting point and extended by additional domain-specific criteria.

- **Dictionary existence:** whether the word can actually be found in WordNet.
- **Consistent naming:** check if the naming style is consistent, e.g., only upper case, initial upper case, lower case, camel case, with or without blanks and/or hyphens.
- **Hypernyms:** if the word has hypernyms in the schema.
- **Synonyms:** if the word has synonyms in the schema.
- **Dates:** if the term “date” occurs in an attribute name, its data type must be `dateTime`.
- **Identifier:** if the term “ID” or “Identifier” occurs in an attribute name, the attribute must be a primary key or at least unique.
- **Foreign keys:** the naming of a foreign key and its corresponding primary key must be equivalent.

Based on these criteria, we suggest calculating a readability score according to

$$Red(s) = \frac{\sum_{i=1}^{|w|} \#fcrit_i / \#crit}{|w|}, \quad (16)$$

where $|w|$ is the total number of words considered, $\#crit$ is the number of considered criteria, and $\#fcrit_i$ is the number of fulfilled criteria per word w_i .

H. Estimation of Integrated Quality Values

In Big Data applications there is usually no gold standard for the entire data set, which makes it impossible to calculate DQ metrics that require a GS in the formula. However, there exist often reference data sets of good quality, for example, purchased customer addresses or a manually cleaned subset of the data. In such cases, DQ can be estimated by extrapolating exact measurements for parts of the data to the entire data set. An estimated quality rating allows to draw conclusions whether to include a data source in a query result or not.

QuaIle provides a heuristic estimation of DQ values for a number of query results, views, and integrated record sets. Assuming a composite record set can be defined by applying only relational algebra operators (projection π , selection σ , rename ρ , union \cup , set difference $-$, and cross product \times [53]) to existing data, queries can be treated as relational syntax trees. From these trees, estimations about the DQ metrics of the composite set can be made without actually evaluating DQ again. Hence, a gold standard is only required for the exact measurement of the leaf components and the DQ estimation for larger (integrated) data is possible without further need of a gold standard [37]. Currently, estimates for the DQ dimensions accuracy, completeness, and pertinence have been implemented in QuaIle. The DQ metrics of the composite set are estimated by traversing the relational algebra syntax tree in a bottom up fashion utilizing the formulas we present in Tables II and III. Here, $D(c)$ is the proportion of records in a data set c , for which at least one duplicate entry exists in c , and p is a selection-specific factor denoting $\frac{|\text{selected records}|}{|\text{original records}|}$.

IV. IMPLEMENTATION ARCHITECTURE AND DEMONSTRATION

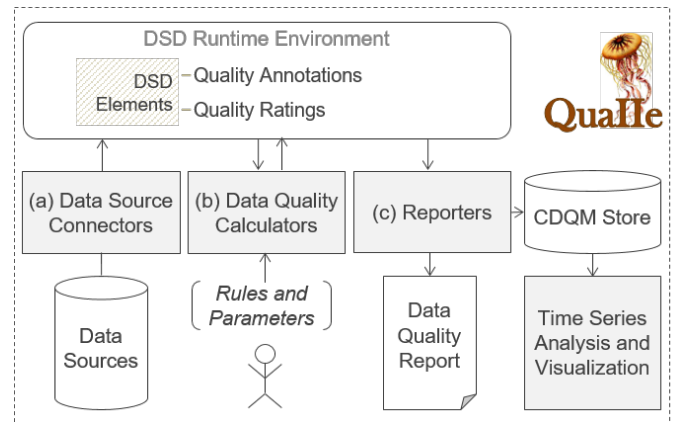


Fig. 1. Implementation Architecture of QuaIle

Fig. 1 shows the architecture of our modular Java-based tool QuaIle for measuring IIS data-level and schema-level quality ad hoc and continuously. The tool consists of three abstract components: (a) data source connectors to establish an IS connection and to load schema information in form of DSD elements, (b) quality calculators that perform schema and data

TABLE II
DATA QUALITY ESTIMATION - COMPLETENESS AND PERTINENCE

Operator	Composite	Completeness of Composite	Pertinence of Composite
Projection	$\pi(c)$	$Com(c)$	$Per(c)$
Selection	$\sigma(c)$	$p * Com(c)$	$Per(c)$
Union	$c_1 \cup c_2$	$\frac{Com(c_1) + Com(c_2) - D(c_1 \cup c_2)}{Com(c_1) + Com(c_2)} * 2$	$\frac{Per(c_1) * c_1 + Per(c_2) * c_2 }{ c_1 + c_2 }$
Set Difference	$c_1 - c_2$	$Com(c_1) - D(c_1 \cup c_2) * \frac{Com(c_1) + Com(c_2)}{2}$	$\frac{2 * Per(c_1) * c_1 - D(c_1 \cup c_2) * (Per(c_1) * c_1 + Per(c_2) * c_2)}{2 * c_1 - D(c_1 \cup c_2) * (c_1 + c_2)}$
Cross Product	$c_1 \times c_2$	$Com(c_1) * Com(c_2)$	$Per(c_1) * Per(c_2)$

TABLE III
DATA QUALITY ESTIMATION - ACCURACY

Operator	Composite	Accuracy of Composite
Projection	$\pi(c)$	$Acc(c)$
Selection	$\sigma(c)$	$\frac{Com(c) * p * Acc(c)}{Com(c) * p + (1 - p) * Acc(c)}$
Union	$c_1 \cup c_2$	$\left(1 - \frac{D(c_1 \cup c_2)}{2}\right) * (Com(c_1) + Com(c_2))$
Set Difference	$c_1 - c_2$	$\frac{2 * Com(c_1) - D(c_1 \cup c_2) * (Com(c_1) + Com(c_2))}{2 * \frac{Com(c_1)}{Acc(c_1)} - D(c_1 \cup c_2) * \left(\frac{Com(c_1)}{Acc(c_1)} - \frac{Com(c_2)}{Acc(c_2)}\right)}$
Cross Product	$c_1 \times c_2$	$\frac{Com(c_1) * Com(c_2)}{1 + Com(c_1) * \left(\frac{Com(c_2)}{Acc(c_2)} - 1\right) + Com(c_2) * \left(\frac{Com(c_1)}{Acc(c_1)} - 1\right) + \left(\frac{Com(c_1)}{Acc(c_1)} - 1\right) * \left(\frac{Com(c_2)}{Acc(c_2)} - 1\right)}$

quality measurement and annotate this information to the DSD elements, and (c) reporters, which either generate a human- and machine-readable quality report, or persist the continuous DQ measurements in a DB for later analysis. The tool has been implemented with a focus on maximum flexibility and extensibility, which makes it easy to add new connectors, calculators, or reporters, due to a standardized interface for each component.

In addition to a pre-configured automatic execution, QuaIle also allows user input in form of rules and parameters for specific quality calculations. The DSD runtime environment, which is described in more detail in Section IV-B, holds schema information on the loaded data sources in form of DSD elements along with assigned quality ratings and annotations. The DSD runtime environment exists only during the Java runtime, that is, during the execution of one (initial) DQ measurement procedure. In order to persist the DQ measurements for later analysis, they need to be exported to the CDQM store using one of the reporters. In the following paragraphs, the selection of data sources used for our demonstration is justified and explained, and each component (a), (b), and (c), as well as the DSD runtime environment and the CDQM store are described in more detail.

A. Demonstration Data Sources

Recently, a call for more empiricism in DQ research has been made in [58], promoting (1) the evaluation on synthetic data sets to show the reproducibility of the measurements, and (2) evaluations on large real-world data sets. In order to cover both requirements, we used curated demo data sources of known quality, which allow manual verification of quality measurements, as well as real-world data sources of unknown quality. We selected those data sources also in a way to demonstrate the usage of all three volatility types (V1-3), from highly volatile, to completely static. All six used data sets are described in the following paragraphs in more detail.

a) *Employees*: The “employees” DB contains six tables with about three million records in total (300,024 in the table “employees” alone) and models the administrations of employees in a company [59]. Since it is a curated demo DB of known quality, we load the original employees DB into the *Datasource* object *dsEmpGS*, which is used as gold standard for our demonstration. In addition, we created two variants that have been automatically populated with randomly inserted errors in the original data: *dsEmp1* (501 records from the main table “employees”) and *dsEmp2* (4,389 records). Table IV shows the error types that were used in the script. The added noise *n* is an absolute error that is normally distributed.

b) *Sakila*: The Sakila DB has 16 tables and models the administration of a film distribution [60]. While the employees

TABLE IV
ERROR TYPES

Error type	Domain	Example
LetterSwap	String	"Bernhard" → "Bernhrad"
LetterInsertion	String	"Bernhard" → "Bernnhard"
LetterDeletion	String	"Bernhard" → "Bernhrd"
LetterReplacement	String	"Bernhard" → "Burnhard"
AddedNoise	Numeric	$a \rightarrow a + n$, where $n \sim N(0, 1)$
NullFault	Any	"Bernhard" → NULL
RecordDuplication	Record	$\{("Werth", 9)\} \rightarrow \{("Werth", 9), ("Werth", 9)\}$
RecordDeletion	Record	$\{("Werth", 9)\} \rightarrow \emptyset$
RecordInsertion	Record	$\{("Werth", 9)\} \rightarrow \{("Werth", 9), ("Ehrlinger", 5)\}$
RecordCrossOver	Record	$\{("Werth", 9), ("Wöß", 2)\} \rightarrow \{("Werth", 2), ("Wöß", 9)\}$

DB contains a large number of records for quality measurement on the data-level, Sakila consists of a more advanced schema and is thus used for schema quality measurement that requires a GS (again, because of the known quality of the DB). We employed the `Datasource` object *dsSakilaGS*, which represents the original Sakila DB, as GS. In addition, we created three modified versions of the Sakila DB: one with minor modifications, but without removing or adding elements to affect correctness (*dsSakila1*), a second one where attributes and tables have been removed to affect completeness (*dsSakila2*), and a third one where attributes and tables have been added to affect pertinence (*dsSakila3*).

c) *Northwind*: The well-known Northwind DB from Microsoft (*dsNorthwind*) can be downloaded from [61] and demonstrates a comprehensive company DB, including 12 tables for e.g., customers, employees, products, order details. In addition to the original Northwind DB, we use an updated version published by dofactory [62] (*dsNorthwindNew*) with more recent dates, but only five tables.

d) *CD CSV*: We also used a CSV file (*dsCD*), which contains real-world music CD data that was originally published on the repeatability website by Felix Naumann [63]. The dataset contains information on the artist, title, genre, year, and contained tracks of 9,763 records, including 299 duplicates and was randomly extracted from freeDB.

e) *Metadynea*: To demonstrate data model independence of schemas investigated with QuaIle, we also employed a Cassandra DB in productive use called "metadynea" (*dsMetadynea*). It contains five column families (tables) with about 60 GB of chemometrics data, which is distributed on three nodes.

f) *Stock Exchange Data*: In order to demonstrate the quality measurement of highly volatile data sets, we used real stock data (open, high, low, close, and volume) since the year 2000 from the equities of IBM, Microsoft, and Apple, accessed through the Alphavantage API (Application Programming Interface) [64]. The connected and analyzed data set is called *dsStockdata*.

As supplement to the demonstration in this paper, we published an executable (*QuaIle.jar*) on our project web-

site [65], which allows to reconstruct the schema quality measurement described in this section. The program takes one mandatory and one optional command line parameter: (1) the path to the DSD schema to be observed and (2) the path to the GS schema, and generates a quality report in XML format. Schema descriptions for all four versions of the Sakila DB, as well as a description for the employees DB are provided in form of DSD files.

B. Data Source Connectors and DSD Environment

A connector's task is to guarantee data model independence by accessing a data source and transforming its schema into a harmonized schema description, which is based on the DSD vocabulary. The transformation process from various data models and details of the DSD vocabulary are described in [21]. The transformation from schema elements to DSD elements is a prerequisite for performing cross-schema calculations and obtaining information about a schema's similarity to other schemas in the IIS. In QuaIle, DSD elements are represented as dynamically created objects in the Java runtime environment. Below we list the most important terms of the DSD vocabulary that are used in this paper.

- A `Datasource s` represents one schema in an IIS and has a type (e.g., relational DB, spreadsheet) and an arbitrary number of concepts and associations, which are also referred to as schema elements.
- A `Concept c` is a real-world object type and is usually equivalent to a table in a relational DB or a class in an object-oriented DB.
- An `Association` is a relationship between two or more concepts. There are three types of association: (i) a reference association describes a general relationship between two concepts (e.g., employment of a person with a company); (ii) an inheritance association represents an inheritance hierarchy (e.g., specific types of employees are inherited from a general employee concept); and (iii) an aggregation association describes the composition of several concepts (components) to an aggregate.
- An `Attribute` is a property of a concept or an association; for example, the column "first_name" provides information about the concept "employees".

Fig. 2 shows an example of a transformation of two relations from the employees DB: `employees {emp_no: int, birth_date: date, first_name: string, last_name: string}` and `dept_emp {emp_no: int, dept_no: int, from_date: date, to_date: date}` into a DSD file in Turtle syntax (cf. [66]). The attribute descriptions are omitted for brevity. The example shows that a relational table can be transformed into a concept or an association, for example, `dept_emp` is a reference association since it models the assignment of an employee to a department.

While this harmonization step enables comparability and quality measurement of schemas from different data models, it

```

1 ex:employees a dsd:Concept ;
2   rdfs:label "employees" ;
3   dsd:hasAttribute ex:employees.emp_no, ex:employees
    .birth_date, ex:employees.first_name, ex:
    employees.last_name;
4   dsd:hasPrimaryKey ex:employees.pk .
5
6 ex:dept_emp a dsd:ReferenceAssociation ;
7   rdfs:label "dept_emp" ;
8   dsd:hasAttribute ex:dept_emp.emp_no, ex:dept_emp.
    dept_no, ex:dept_emp.from_date, ex:dept_emp.
    to_date;
9   dsd:hasPrimaryKey ex:dept_emp.pk ;
10  dsd:referencesTo ex:employees, ex:departments .

```

Fig. 2. Example Schema Description

does not guarantee access to the original information sources' content after transformation. Consequently, the schema quality metrics in QuaIle primarily use the schema's metadata instead of the IS content. An exception is the determination of the normal form, which is impossible without considering the semantics of the attributes that can be derived from the content.

There are two different types of connectors in QuaIle: (1) data source connectors (DSConnector), which load the meta data of an IS to describe its schema, and instance connectors (DSInstanceConnector), which additionally provide access to the data values of an IS. The interface-oriented design of QuaIle allows new connectors to be added by implementing one of the two abstract classes DSConnector or DSInstanceConnector. Currently, five different connectors are supported:

- ConnectorMySQL creates a connection to a MySQL DB as representative for relational DBs by using the functionality of the MySQL Java Connector (cf. [67]). This connector allows access to the DB data values. Information on the selected DB schema is retrieved from the data dictionary, including all tables, columns, foreign keys and column properties.
- ConnectorCSV allows to access CSV files and is also a subclass of DSInstanceConnector. Due to little meta data that is available in plain CSV files, schema information is solely extracted from the given file (i.e., column headers as attribute names).
- ConnectorOntology uses the Apache Jena framework (cf. [68]) to access a DSD file. Since DSD files hold only schema information and not a connection to the original database, this connector does not provide any possibilities for accessing the DB content and can be used for schema quality measurement only.
- ConnectorCassandra uses the Datastax Java driver (cf. [69]) to access a Cassandra DB. This connector currently operates on schema-level only to evaluate the DQ measurements not only on relational data, but also on denormalized wide-column-store schemas.
- ConnectorAlphavantage is a domain-specific connector to crawl and load stock market data from the Alphavantage API [64].

Fig. 3 shows an example instantiation for each of the

domain-independent connector types. In addition to opening a connection, it is necessary to load the schema and thus trigger the conversion of schema elements to DSD elements in the Java DSD environment. For our demonstration, we created a connection to all data sources described in Section IV-A, adhering to the same naming standard. For example, for the employees DB we created the connectors *connEmp1* and *connEmp2* to access the MySQL databases with the inserted errors, and load their schema in form of two Datasource objects *dsEmp1* and *dsEmp2* into the DSD environment.

```

1 // Opening and loading a MySQL data source
2 DSInstanceConnector connEmpGS = ConnectorMySQL.
    getInstance("jdbc:mysql://localhost:3306/", "
    employees", "user", "pw");
3 Datasource dsEmpGS = connEmpGS.loadSchema();
4
5 // Opening and loading a DSD schema description
6 DSConnector connSakilaGS = new ConnectorOntology("
    filepath/sakila_gs.ttl", "Sakila_Goldstandard");
7 Datasource dsSakilaGS = connSakilaGS.loadSchema();
8
9 // Opening and loading a CSV file
10 DSInstanceConnector connCD = new ConnectorCSV("
    filepath/cd.csv", ",", "\n", "CD");
11 Datasource dsCD = connCD.loadSchema();
12
13 // Opening and loading a Cassandra data source
14 DSConnector connMeta = new ConnectorCassandra("
    hostname", "metadynea", "user", "pw");
15 Datasource dsMeta = connMeta.loadSchema();

```

Fig. 3. Data Source Connectors

In QuaIle, each data source connector also offers at least one gold standard implementation, in order to allow the calculation of reference-based DQ dimensions (e.g., completeness). Fig. 4 shows the creation of two different gold standards: (1) *empGS*, which can be used for quality measurement at the data-level, and (2) *sakGS1*, a *DSDSchemaGS* that is solely used for schema quality measurement. Since specific gold standard implementations might have different tasks, each implementation requires a different set of parameters. However, all gold standards in QuaIle inherit from the abstract class *GoldStandard*, which offers methods to retrieve referenced records or schema elements. The object *empGS* in Fig. 4 shows the instantiation of a gold standard object for a single concept (table), for DQ calculations on different aggregation levels (i.e., when only parts of the content of a data source should be analyzed).

The *DSDSchemaGS* for schema quality calculations extends the idea of simply representing a perfect reference to an information source; rather it is a “container” that holds the reference to another information source and calculates the similarity or dissimilarity between schema elements on-the-fly. Thus, it is, for example, possible to compare one MySQL DB schema to a DSD description as shown in Fig. 4, to overcome data model heterogeneity.

C. Data Quality Calculators

Each DQ calculator is dedicated to one of the quality dimensions described in Section III and links the measurements to

```

1 // Creation of a gold standard from a single concept
2 GoldStandard empGS = new StrictConceptMySQLGS(
    dsEmpGS.getConcept("employees"), connEmpGS);
3
4 // Creation of a schema gold standard
5 GoldStandard sakGS1 = new DSDSchemaGS(dsSakila1,
    dsSakilaGS);

```

Fig. 4. Gold Standards

the corresponding DSD elements in the DSD runtime environment. Quality measurements in the DSD runtime environment can be used for reporting or reused by other calculators, and can be divided into two different types: *quality ratings* or *quality annotations*. A rating is a double value between 0.0 and 1.0, which is calculated by a specific metric that is assigned to a quality dimension. An example for a DQ rating is a value of 0.85 for the dimension “completeness” on data-level using the metric “ratio”. A quality annotation can be an arbitrary object that is linked to a DSD element in order to provide additional information about the quality. An example would be the annotation of functional dependencies to a concept. In the following, we summarize all DQ calculators that are currently implemented in QuaIle and link them to the respective metrics from Section III, grouped by dimension:

- Accuracy / Correctness
 - RefCorrectnessCalculator (1) - data
 - RatioAccuracyCalculator (2) - data
 - DSDCorrectnessCalculator (3) - schema
- Completeness
 - RatioCompletenessCalculator (4) - data
 - UniqueRatioCompletenessCalculator (6) - data
 - FilledCalculator (7) - data
 - DSDCompletenessCalculator (8) - schema
- Pertinence
 - RatioPertinenceCalculator (9) - data
 - DSDPertinenceCalculator (10) - schema
- Timeliness
 - AverageCurrencyCalculator (12) - data
 - AverageTimelinessCalculator (13) - data
- Minimality / Duplicity
 - RecordMinimalityCalculator (14) - data
 - SchemaMinimalityCalculator (15) - schema
- Readability
 - SchemaReadabilityCalculator (16) - schema
- Normalform
 - NormalFormCalculator - schema

Fig. 5 shows the application of all non-time-related DQ calculators that are implemented in the current version of QuaIle. Initially, the concept “employees” from the erroneous Datasource *dsEmp1* is selected for closer investigation. As an example for a distance function, which is required for the minimality calculation, line 5-7 cover the creation of an *EnsembleDistance*, which is a weighted combination

of an arbitrary number of specific distance functions. In the demonstration, we use a combination of two string distances for the attributes *first_name* and *last_name* in the “employees” table. However, QuaIle allows the creation of arbitrary complex distance functions for each record. Finally, ratings for the DQ dimensions accuracy, completeness, pertinence, and minimality are calculated. Line 16 shows how to programmatically retrieve those stored DQ values from the DSD runtime environment. One data quality rating or annotation is uniquely identifiable by a reference to the DSD element (e.g., a reference to the concept “employees” in *dsEmp1*), the *DIMENSION_LABEL* of the measured quality dimension (e.g., “completeness”) as well as a *METRIC_LABEL* (e.g., “ratio”), which describes the metric used for calculating the dimension.

```

1 // Select concept "employees" from employees DB
2 Concept c = dsEmp1.getConcept("employees");
3
4 // Create a custom distance measure
5 EnsembleDistance<Record> dist = new EnsembleDistance
    <Record>();
6 dist.addDistance(new StringRecordDistance(c,
    getAttribute("first_name"), new
    LevenshteinDistance(), 0.5);
7 dist.addDistance(new StringRecordDistance(c,
    getAttribute("last_name"), new
    LevenshteinDistance(), 0.5);
8
9 // Perform quality calculations
10 RatioAccuracyCalculator.calculate(c, empGS, connEmp1);
11 RatioCompletenessCalculator.calculate(c, empGS,
    connEmp1);
12 RatioPertinenceCalculator.calculate(c, empGS,
    connEmp1);
13 RecordMinimalityCalculator.calculate(c, dist, 0.1,
    connEmp1);
14
15 // Retrieve DQ measurements from the DSD runtime
    environment (formerly "Data Quality Store")
16 DataQualityStore.getDQValue(c,
    RatioPertinenceCalculator.DIMENSION_LABEL,
    RatioPertinenceCalculator.METRIC_LABEL)

```

Fig. 5. Data Quality Calculations

In addition to the measurement of *dsEmp1* (501 records), we applied the same calculations on the “employees” table of *dsEmp2* (4,389 records). The results can be compared in Table V. The low quality values for accuracy and completeness result from the small subsets of the erroneous tables in contrast to the original employees table with 300,024 records.

TABLE V
DQ MEASUREMENT OF ERRONEOUS DATA SOURCES

Dimension	Metric	<i>dsEmp1</i>	<i>dsEmp2</i>
Accuracy	Ratio	0.0013	0.0116
Completeness	Ratio	0.0013	0.0116
Pertinence	Ratio	0.7725	0.7938
Minimality	Record	0.7180	0.7532

The time-related DQ dimensions currency and timeliness require information about the last update of a tuple, which is not available in the employees DB, but offered by Sakila DB in

form of an attribute `last_update`. We show the calculation in Fig. 6, which leads to a rating of 3.7156 for currency and 0.9988 for timeliness. Both calculators require information about the attribute that holds the update information. In addition, the timeliness calculation requires a parameter for the volatility. In the demonstration, the volatility is set to a value of 10 years for actor names, since it is very unusual that a name for an actor changes within that time frame.

We want to point out that due to their definition and in contrast to the other DQ dimensions, it is not possible to assess the time-related dimensions without prior knowledge, which does not align with the objective of QuaIIE to be domain-independent. However, we included those two DQ calculators in order to provide comprehensive DQ measurement and think it is worth investigating both dimensions in more detail in order to come up with automatically suggested parameters (e.g., for the volatility).

```

1 Concept actor = dsSakila1.getConcept("actor");
2 double volatility = 10 * MILLIS_IN_SEC * SEC_IN_MIN
  * MIN_IN_HOUR * HOUR_IN_DAY * DAY_IN_YEAR;
3
4 AverageCurrencyCalculator.calculate(connSakila1,
  actor, r -> (Date) r.getField("last_update"));
5 AverageTimelinessCalculator.calculate(connSakila1,
  actor, r -> (Date) r.getField("last_update"),
  volatility);

```

Fig. 6. Time-Related Data Quality Calculations

For the schema quality calculations, we employed a DSD description of the original Sakila DB as gold standard and accessed the three additional data sources (*dsSakila1*, *dsSakila2*, *dsSakila3*) through the MySQL connector. Each data source contains schema modifications that tackle one of the schema quality dimensions correctness, completeness, and pertinence, and are justified in the following paragraphs. For the demonstration using QuaIIE.jar on our project website [65], we provided all four schemas as DSD files in order to facilitate data exchange and reproduction.

The 16 tables from Sakila were transformed into 14 DSD concepts and two DSD reference associations (`film_category` and `film_actor`). For the *DSD similarity*, standard parameters have been used with a less restrictive attribute similarity threshold of 0.8. The determination and evaluation of the schema similarity standard parameters is explained in [36]. Fig. 7 shows the application of the schema quality calculators correctness, completeness, pertinence, minimality, and normalization.

```

1 DSDCorrectnessCalculator.calculate(dsSakila1, sakGS1);
2 DSDCompletenessCalculator.calculate(dsSakila2, sakGS2);
3 DSDPertinenceCalculator.calculate(dsSakila3, sakGS3);
4 RatioMinimalityCalculator.calculate(dsSakilaGS);
5 NormalFormCalculator.calculate(dsSakilaGS, connSakilaGS);

```

Fig. 7. Schema Quality Calculations

The results of the schema quality measurements applied to Sakila DB, employees DB, stock data, Northwind DB and

Metadynea DB are provided in Table VI. The results are discussed in more detail in the following subsections.

TABLE VI
SCHEMA QUALITY MEASUREMENT RESULTS

Schema	Cor	Com	Pert
<i>dsSakilaGS</i>	1.0	1.0	1.0
<i>dsSakila1</i>	0.813	1.0	1.00
<i>dsSakila2</i>	0.929	0.813	0.929
<i>dsSakila3</i>	0.824	0.938	0.882
	Min	NF (t.=tables)	
<i>dsSakilaGS</i>	1.0	6 t.: BCNF, 9 t.: 2NF, 1 t.: 1NF	
<i>dsEmpGS</i>	0.8	All BCNF	
<i>dsStockdata</i>	1.0	1 t.: BCNF	
<i>dsNorthwind</i>	1.0	6 t.: BCNF, 4 t.: 2NF, 1 t.: 1NF	
<i>dsMetadynea</i>	0.667	Not applicable for Cassandra	

a) *Schema Correctness*: In order to demonstrate the correctness dimension, we performed changes in the observed schema but did not remove or add new schema elements. The corresponding DQ report can be generated by executing `java -jar QuaIIE.jar sakila_correctness.ttl sakila_gs.ttl`. First, the concept `film` was renamed to “movie”, which did not change the ratings for pertinence and completeness, but decreased correctness slightly to 0.938 due to the additional incorrect element. Second, all occurrences of `film_id` in the DB were replaced with “movie”. While completeness and pertinence retained a rating of 1.0, because all concepts and associations were assigned (even if incorrectly) to their original correspondences in the GS, correctness achieved only a rating of $\frac{13}{13+3+0} = 0.813$.

b) *Schema Completeness*: The completeness calculation was performed by removing schema elements. The DQ report for this demonstration can be generated by assessing `sakila_completeness.ttl`. Initially, the two tables `category` and `film_category` were removed, which resulted in a completeness rating of $\frac{14+0}{14+0+2} = 0.875$ because two elements were classified as missing. Then, the attribute `picture` was deleted from the table `staff`. Removals at the attribute-level did not directly affect the result of the completeness calculation, since `staff` is still correctly assigned to its gold standard representation due to the tolerance of the distance calculation. Concluding, three additional attributes were removed from `staff`, which resulted in a similarity rating of 0.692 between `staff` and its correspondence in the GS. Consequently, both tables were not mapped because they were too different and completeness dropped to $\frac{13+0}{13+0+3} = 0.813$.

c) *Schema Pertinence*: For the demonstration of pertinence, we added additional elements to the schema and the quality report can be generated by assessing the file `sakila_pertinence.ttl`. In a first step, the “employees” table from the employees DB was added to *dsSakila3*, dropping pertinence to 0.941. This demo correctly classifies the concept `employees` as an extra element, although the new concept has a relatively low distance to the concept

actor. Second, we modified the concept *actor* in *dsSakila3*, such that no assignment to its corresponding concept in the GS was created and the pertinence rating dropped to $\frac{15+0}{15+0+2} = 0.882$. Following, the newly added *employees* table was aligned with the *actor* concept in the GS by removing and altering attributes. This resulted in a similarity value of 0.833 between *employees* and the concept *actor* from the GS and increased completeness to 1.0 (all elements could be assigned to the GS). However, the pertinence dimension (0.941) indicated the extra *actor* concept in the observed schema, which did not match any of the GS elements.

We conclude that an examination of all three dimensions (correctness, completeness, and pertinence) is advisable when measuring the quality of a schema. Note that the correctness metric is particularly strict, because it is decreased by every incorrect element in the schema, whereas completeness and pertinence do not distinguish between correct and incorrect.

The results of all four schema quality measurement results are summarized in Table VI and elaborated in more detail in the following paragraphs.

d) Schema Minimality: Analogous to the data minimality, schema minimality requires a distance function. Currently, two schema distance functions are offered: the similarity flooding algorithm introduced in [52] and DSD similarity, which we use in the following calculations with standard parameters that have been evaluated in [36]. The schemas of the Sakila DB (*sakila_gs.ttl*), the Northwind DB and stock data achieve an ideal minimality rating of 1.0, because all schema elements are sufficiently different to each other. For the Sakila DB with 16 schema elements, minimality is calculated according to $\frac{16-1}{16-1} = 1.0$.

However, the minimality ratings of the *employees* DB and *metadynea* are clearly below 1.0. This rating is expected for a Cassandra DB schema like *metadynea*, where denormalization (and thus, redundancy at the schema-level) is an intended design decision. In order to investigate the reason behind the minimality rating for the *employees* schema in more detail, we observe the similarity matrix from the DSD similarity in Table VII. Interestingly, the two associations *dept_emp* and *dept_manager* achieve a very high similarity of 0.875, which reduces the minimality rating to $\frac{5-1}{6-1} = 0.8$. In practice, this rating indicates an IS architect that the two associations should be further analyzed. However, in our case, no further action is required since the *employees* schema contains a special modeling concept of parallel associations (i.e., two different roles), which does not represent semantic redundancy, but leads to very similar relations in the schema model (cf. [59]). Since it is known that this modeling construct yields high similarity values (e.g., also for schema matching applications), it was specially suited to demonstrate our minimality metric. The full quality report for this demo can be generated by executing “java -jar QuaIIE.jar *employees.ttl*”.

e) Normal Form Calculation: The NF calculator was applied to the *employees* DB and yields BCNF for each

TABLE VII
SIMILARITY MATRIX FOR EMPLOYEES SCHEMA

	depts*	dept_emp	dept_mgr*	employees	salaries	titles
depts*	1.0	0.125	0.125	0.1	0.125	0.125
dept_emp	0.125	1.0	0.875	0.1818	0.2222	0.1
dept_mgr*	0.125	0.875	1.0	0.1818	0.2222	0.1
employees	0.1	0.1818	0.1818	1.0	0.1818	0.1818
salaries	0.125	0.2222	0.2222	0.1818	1.0	0.375
titles	0.125	0.1	0.1	0.1818	0.375	1.0

*Departments is abbreviated with “depts” and dept_manager with “dept_mgr”.

concept. The minimal cover of the FDs is shown in Table VIII. Due to the considerable number of records in the *employees* database, calculating these results took about 22 minutes and 45 seconds on a Macbook Pro with an Intel Core i7 processor with 2.2 GHz and 16 GB main memory. In addition to FDs, candidate keys are also annotated to the observed schema elements, and attributes are annotated with a Boolean value that indicates whether they are classified as key or non-key. Note that, particularly in terms of performance, more sophisticated methods of discovering FDs exist [54]. However, since the main aim of our work was to provide a comprehensive approach to data and schema quality measurement, the normalization dimension was included to support full FD discovery (i.e., without approximation).

TABLE VIII
NF CALCULATION - EMPLOYEES SCHEMA

Concept	Functional Dependencies
departments	{dept_no} → {dept_name}, {dept_name} → {dept_no}
dept_emp	{emp_no, dept_no} → {from_date, to_date}
dept_manager	{emp_no} → {dept_no, from_date, to_date}
employees	{emp_no} → {first_name, last_name, gender, birth_date, hire_date}
salaries	{emp_no, from_date} → {to_date, salary}
titles	{emp_no, title, from_date} → {to_date}

D. Data Source Integration

In IIS, it is often necessary to estimate the quality of data stemming from different IS. QuaIIE supports the virtual integration of different concepts, which is realized with the Java classes *IntegratedDatasource* and *IntegratedConcept*. Fig. 8 shows an example integration, where all records from the table “employees”, which is present in both erroneous data sources *dsEmp1* and *dsEmp2*, are unified. The data is stored in form of a virtual integrated data source (*ids*), which exists only during runtime.

An integrated concept contains an *operator tree*, which specifies the data sources, concepts, connectors, and integration transformations that are required for its creation. After generating such an integrated concept, it can be assessed likewise to an ordinary concept from a single data source in QuaIIE (cf. lines 3-6 in Fig. 9). Additionally, it is possible to estimate the quality (cf. Section III-H), which is not a complete measurement of the new integrated concept, but

```

1 IntegratedDatasource ids = DSDFactory.
  makeIntegratedDatasource("integratedEmp");
2
3 ISQLIntegrator integrator = new ISQLIntegrator(ids);
4 integrator.add(dsEmp1, connEmp1);
5 integrator.add(dsEmp2, connEmp2);
6
7 IntegratedConcept ic = integrator.
  makeIntegratedConceptFromString("SELECT * FROM
  dsEmp1.employees UNION SELECT * FROM dsEmp2.
  employees", "integratedEmployees");

```

Fig. 8. Data Integration

is based on the prior quality ratings of each IS. Thus, an estimation requires the prior measurement of each IS that takes part in the integration.

```

1 DSInstanceConnector integrConn = new
  IntegratedInstanceConnector(ic);
2
3 RatioCompletenessCalculator.calculate(ic, gsEmp,
  integrConn);
4 RatioAccuracyCalculator.calculate(ic, gsEmp,
  integrConn);
5 RatioPertinenceCalculator.calculate(ic, gsEmp,
  integrConn);
6
7 RatioCompletenessCalculator.estimate(ic);
8 RatioAccuracyCalculator.estimate(ic);
9 RatioPertinenceCalculator.estimate(ic);

```

Fig. 9. DQ Estimation of an Integrated Concept

The ratings for the DQ calculations and estimations from Fig. 9 are compared in Table IX and show high conformance. In the current version of QuaIle, quality estimation is only available for the dimensions accuracy, completeness, and pertinence. However, an extension of the DQ estimators to other dimensions, like minimality, is planned as future work.

TABLE IX
DQ CALCULATION OF AN INTEGRATED CONCEPT

Dimension	Metric	Measurement	Estimation
Accuracy	Ratio	0.0129	0.0130
Completeness	Ratio	0.0129	0.0128
Pertinence	Ratio	0.7916	0.7916

E. Data Quality Reports

In order to present the quality ratings and annotations contained in the DSD runtime environment in a human- and machine-readable way, QuaIle offers several reporter classes that generate a quality report. The most comprehensible end-user report is XMLTreeStructureDQReporter, which is created in Fig. 10 and exports a description of all connected data sources with their DSD elements, quality ratings and annotations. Since such a report tends to be large and verbose for large IIS, the hierarchical structure of the XML document allows to drill-down and roll-up on different aggregation levels by using a suitable viewer. In addition, languages like XSLT, XQuery, or XPath allow a user to search within such a report. The advantage of an XML report in our use case is the

fact that it can be reused automatically for further analysis and benchmarking (e.g., for data quality monitoring). When measuring the quality of the published DSD schemas with QuaIle.jar (cf. [65]), the output is such a report.

```

1 XMLTreeStructureDQReporter reporter = new
  XMLTreeStructureDQReporter();
2 reporter.buildReport();
3 reporter.writeReport("path/DQReport.xml");

```

Fig. 10. Data Quality Report Generation

The exemplary quality report in Fig. 11 shows an excerpt of the assessed data source *dsEmp1*, which illustrates possible quality ratings and annotations on schema (here: completeness, minimality, and normalization) and on the data-level (here: accuracy and pertinence).

```

1 <DataQualityReport>
2 <Datasource label="employees" URI="http://example.
  com/dsEmp1">
3 <Quality>
4 <Ratings>
5 <Completeness>
6 <DSDSchema>1.0</DSDSchema>
7 </Completeness>
8 <Minimality>
9 <Ratio>0.8</Ratio>
10 </Minimality>
11 </Ratings>
12 </Quality>
13 <Concept label="employees" URI="http://example.com
  /dsEmp1/employees">
14 <Quality>
15 <Ratings>
16 <Accuracy>
17 <Ratio>0.0013</Ratio>
18 </Accuracy>
19 <Pertinence>
20 <Ratio>0.7725</Ratio>
21 </Pertinence>
22 </Ratings>
23 <Annotations>
24 <Candidate_Key>[{emp_no}]</Candidate_Key>
25 <Normal_Form>BCNF</Normal_Form>
26 </Annotations>
27 </Quality>
28 <Attribute label="emp_no" URI="http://example.
  com/dsEmp1/employees/emp_no">
29 <Quality>
30 <Ratings>
31 <Key_Attribute>
32 <Pseudo_Boolean>1.0</Pseudo_Boolean>
33 </Key_Attribute>
34 </Ratings>
35 </Quality>
36 </Attribute>
37 ...
38 </Concept>
39 ...
40 </Datasource>
41 ...
42 </DataQualityReport>

```

Fig. 11. XML Data Quality Report

The first level below the root node lists all connected data sources that contain quality ratings for the entire Datasource as well as concepts and associations as child nodes. Concepts and associations are further subdivided into their comprising attributes and again, quality ratings and

annotations on concept- or association-level, respectively. Attributes constitute the deepest level in the schema-level hierarchy and can contain quality information on their own, e.g., whether an attribute is a key attribute or not.

F. Continuous Data Quality Measurement

In order to demonstrate the practical application of CDQM with QualLe, we assume the following use cases, ordered by data volatility type: (V1) monitoring the conformance of Wikipedia data to a static gold standard, (V2) measuring the quality of sales data that is loaded on a daily basis with an ETL job into a Data Warehouse, and (V3) continuous measurement of stock data to support data analytics. All three use cases are fictional stories, fueled by real-world data, in order to provide demonstrative examples how CDQM can be applied to different types of data. Fig. 12 depicts the Entity-Relationship diagram of our CDQM store, which follows the suggestion for such a store proposed in [22]. Each CDQM measurement is uniquely identified by the respective DSD element ID (which is an URI), the metric ID, and a timestamp when the measurement was taken. Thus, for a specific CDQM chart, it is only necessary to query for the desired element, metric, and time-range to plot the information for a user.

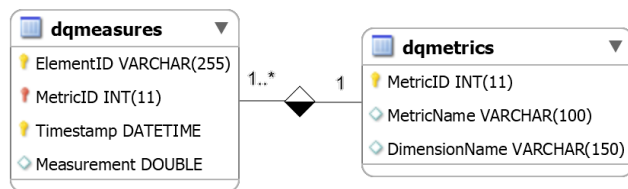


Fig. 12. ER Diagram of the CDQM Store

1) *CDQM for Static Data*: Assume a Wikipedia administrator wants to automatically monitor the state of the articles about Municipal districts in Ireland [70]. The list in [70] provides him with a gold standard in terms of completeness, since it contains all names of Municipal districts in Ireland. This data set is considered static (type V1), because the addition, removal, or merging of Municipal districts is an extremely rare event. In an idealized state (completeness=1.0), there exists a Wikipedia article for each district. The administrator monitors the completeness of these articles between the years 2000 and 2010 and experiences a typical completeness development for static data, illustrated in Fig. 13. The completeness grows strictly monotonous, which is due to the fact that articles are never removed or outdated and the gold standard remains constant (no new Municipal districts arise). This use case also shows that CDQM can be applied to semi-structured data.

2) *CDQM of Incoming Sales Data*: In a large company, sales data (here a derivative of the Northwind DB [61]) is collected every batch-wise from different sales persons to be loaded into the central Data Warehouse (DWH). The DWH administrator wants to monitor the quality of each incoming file and the central DWH. Fig. 14 shows the currency of the

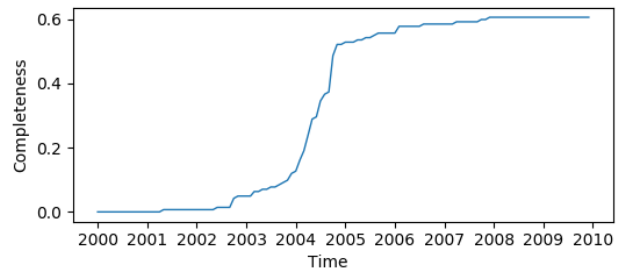


Fig. 13. Completeness of Data extracted from Wikipedia

DWH, which achieves constantly higher values, because the average age of a sale increases as time progresses and the DWH contains more and more older sales. However, intuitively the quality of the DWH is not deteriorating as the sales data does not get less trustworthy. This indicates that currency is probably a too naive measure in the context of accumulating data. In the lower half of Fig. 14, the completeness of the DWH and the average completeness of collected data per day are shown. While the variance of the completeness of the newly arriving data is consistently high, the variance of the completeness for the DWH decreases and almost converges around 0.95.

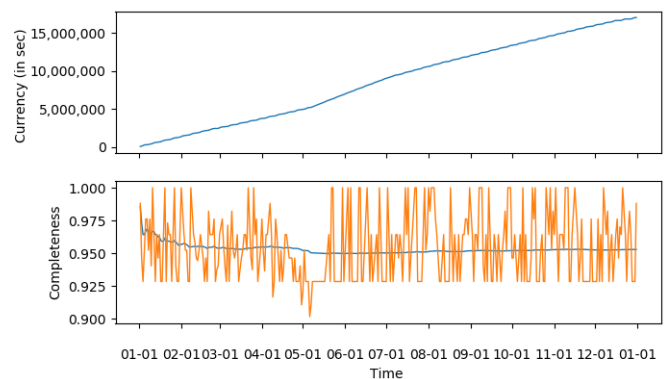


Fig. 14. Currency and Completeness of Sales Data

3) *CDQM for Data Analytics*: Assume, a stock market data analyst creates a machine learning (ML) model that uses the last 50 records of 5-minute-interval stock data, which is of volatility type V3. The model is always built 2 minutes after a new record arrives and predicts the further development for a specific equity (in our example we used IBM). In order to ensure the validity of the model result, we continuously measure the quality of the employed data, each time the model is created. Fig. 15 shows metrics for the DQ dimensions completeness and currency for 10 days between July 10th and 20th, 2018. While the completeness rating remains constantly at 1.0, i.e., all records are constantly delivered, the currency varies. This is an example for intended variation in the quality chart. Since currency describes how up-to-date the data is at hand, and the stock market is closed over the weekend (big

peak) and after 4 pm (small peaks), the model prediction with respect to currency is poor when the market closes and reaches a stable phase during the day. Since during one day only 78 5-minute records are delivered, the data analyst can create only about 28 models where the data has the best currency.

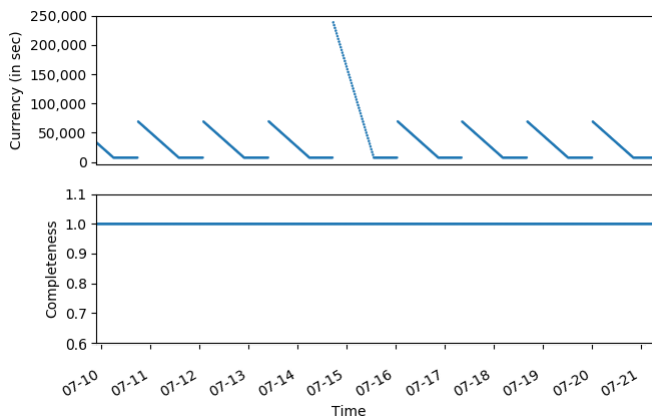


Fig. 15. Currency and Completeness of Stock Data for ML Model

V. CONCLUSION AND FUTURE WORK

In this paper, we have demonstrated how to measure data quality (1) ad hoc and without preparation activities, and (2) continuously over time, using our previously developed DQ tool QuaIle [1]. QuaIle covers metrics for the DQ dimensions accuracy, correctness, completeness, pertinence, timeliness, minimality, readability, and normalization on both, data-level and schema-level of an IS. In addition, it is possible to estimate the quality of integrated IS. To the best of our knowledge, there exists no other DQ tool that implements such a large number of different dimensions and supports ad hoc as well as continuous DQ measurement in a single application. The major contribution in this paper is to highlight the importance for initial ad hoc DQ measurement to get a first insight into DQ, as well as continuous DQ measurement over time, which allows to observe how DQ evolves and to detect unexpected changes in the data.

As has been seen in Section IV-F, the time-progression of some DQ measures can diverge from an intuitive understanding of data quality, e.g., the currency of aggregated data. Consequently, a need for specialized metrics for different types of data in CDQM arises.

Our ongoing work includes long-term evaluations of QuaIle's continuous measurement functionality. In addition, a connector for Oracle DBs and a graphical user interface to visualize the DQ measurements are currently under development. We also plan to implement connectors for other NoSQL DB types like document stores and graph DBs to further evaluate the comparability of DQ measurements on different schema models.

ACKNOWLEDGMENT

The research reported in this paper has been partly supported by the Austrian Ministry for Transport, Innovation and Technology, the Federal Ministry for Digital and Economic Affairs, and the Province of Upper Austria in the frame of the COMET center SCCH. In addition, the authors would like to thank Gudrun Huszar for the implementation of the readability calculator and Julia Hilber for her research work on the Cassandra connector.

REFERENCES

- [1] L. Ehrlinger, B. Werth, and W. Wöß, "QuaIle: A Data Quality Assessment Tool for Integrated Information Systems," in *Proceedings of the Tenth International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA 2018)*. International Academy, Research, and Industry Association (IARIA), 2018, pp. 21–31.
- [2] KPMG International, "Now or Never: 2016 Global CEO Outlook," 2016.
- [3] T. C. Redman, "The Impact of Poor Data Quality on the Typical Enterprise," *Communications of the ACM*, vol. 41, no. 2, Feb. 1998, pp. 79–82.
- [4] T. C. Redman, "Bad Data Costs the U.S. \$3 Trillion Per Year," *Harvard Business Review*, 2016, <https://hbr.org/2016/09/bad-data-costs-the-u-s-3-trillion-per-year> [retrieved: November 2018].
- [5] B. Otto and H. Österle, *Corporate Data Quality: Prerequisite for Successful Business Models*. Berlin, Germany: Springer Gabler, 2016.
- [6] E. E. Ruppert, R. D. Barnes, and R. S. Fox, *Invertebrate Zoology: A Functional Evolutionary Approach*, 7th ed. Cengage Learning, 2003.
- [7] The Apache Software Foundation, "Apache Cassandra," Online, <http://cassandra.apache.org> [retrieved: November 2018].
- [8] D. C. Baulcombe, S. Chapman, and S. Santa Cruz, "Jellyfish Green Fluorescent Protein as a Reporter for Virus Infections," *The Plant Journal*, vol. 7, no. 6, 1995, pp. 1045–1053.
- [9] F. Naumann, U. Leser, and J. C. Freytag, "Quality-driven Integration of Heterogeneous Information Systems," in *Proceedings of the 25th International Conference on Very Large Data Bases*, ser. VLDB '99. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999, pp. 447–458.
- [10] M. Y. Selvege, S. Judah, and A. Jain, "Magic Quadrant for Data Quality Tools," Gartner, Tech. Rep., October 2017.
- [11] J. Barateiro and H. Galhardas, "A Survey of Data Quality Tools," *Datenbank-Spektrum*, vol. 14, no. 15–21, 2005, p. 48.
- [12] V. Pushkarev, H. Neumann, C. Varol, and J. R. Talburt, "An Overview of Open Source Data Quality Tools," in *Proceedings of the 2010 International Conference on Information & Knowledge Engineering, IKE 2010, July 12–15, 2010, Las Vegas Nevada, USA*, 2010, pp. 370–376.
- [13] V. S. V. Pulla, C. Varol, and M. Al, *Open Source Data Quality Tools: Revisited*. Springer International Publishing, 2016, pp. 893–902.
- [14] L. Sebastian-Coleman, *Measuring Data Quality for Ongoing Improvement: A Data Quality Assessment Framework*. Newnes, 2012.
- [15] Y. Wand and R. Y. Wang, "Anchoring Data Quality Dimensions in Ontological Foundations," *Communications of the ACM*, vol. 39, no. 11, Nov. 1996, pp. 86–95.
- [16] N. R. Chrisman, "The Role of Quality Information in the Long-Term Functioning of a Geographic Information System," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 21, no. 2, 1983, pp. 79–88.
- [17] R. Y. Wang and D. M. Strong, "Beyond Accuracy: What Data Quality Means to Data Consumers," *Journal of Management Information Systems*, vol. 12, no. 4, Mar. 1996, pp. 5–33.
- [18] C. Batini and M. Scannapieco, *Data and Information Quality: Concepts, Methodologies and Techniques*. Switzerland: Springer International Publishing, 2016.
- [19] C. Batini, C. Cappiello, C. Francalanci, and A. Maurino, "Methodologies for Data Quality Assessment and Improvement," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, 2009, p. 16.
- [20] A. Maydanchik, *Data Quality Assessment*. Bradley Beach, NJ, USA: Technics Publications, LLC, 2007.

- [21] L. Ehrlinger and W. Wöß, "Semi-Automatically Generated Hybrid Ontologies for Information Integration," in *Joint Proceedings of the Posters and Demos Track of 11th International Conference on Semantic Systems – SEMANTiCS2015 and 1st Workshop on Data Science: Methods, Technology and Applications (DSci15)*. CEUR Workshop Proceedings, 2015, pp. 100–104.
- [22] L. Ehrlinger and W. Wöß, "Automated Data Quality Monitoring," in *Proceedings of the 22nd MIT International Conference on Information Quality (MIT ICIQ 2017)*, J. R. Talburt, Ed., UA Little Rock, Arkansas, USA, 2017, pp. 15.1–15.9.
- [23] B. Heinrich, D. Hristova, M. Klier, A. Schiller, and M. Szubartowicz, "Requirements for Data Quality Metrics," *Journal of Data and Information Quality*, vol. 9, no. 2, Jan. 2018, pp. 12:1–12:32.
- [24] Microsoft Inc., "Task Scheduler," 2018, <https://docs.microsoft.com/en-us/windows/desktop/taskschd/task-scheduler-start-page> [retrieved: November 2018].
- [25] AQR Capital Management, LLC, Lambda Foundry, Inc. and PyData Development Team, "Python Data Analysis Library - pandas," 2018, <https://pandas.pydata.org> [retrieved: November 2018].
- [26] J. Hunter, D. Dale, E. Firing, and M. Droettboom, "Matplotlib," 2018, <https://matplotlib.org> [retrieved: November 2018].
- [27] J. Adelman, M. Baak, N. Boelaert, M. D'Onofrio, J. A. Frost, C. Guyot, M. Hauschild, A. Hoecker, K. J. C. Leney, E. Lytken, M. Martinez-Perez, J. Masik, A. M. Nairz, P. U. E. Onyisi, S. Roe, S. Schaezel, and M. G. Wilson, "ATLAS Offline Data Quality Monitoring," *Journal of Physics: Conference Series*, vol. 219, no. 4, 2010, p. 042018.
- [28] V. Raman and J. M. Hellerstein, "Potter's Wheel: An Interactive Data Cleaning System," in *Proceedings of the 27th VLDB Conference, Roma, Italy*, vol. 1, 2001, pp. 381–390.
- [29] S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer, "Wrangler: Interactive Visual Specification of Data Transformation Scripts," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011, pp. 3363–3372.
- [30] A. Rolland, "MobyDQ," 2018, <https://github.com/mobydq/mobydq> [retrieved: November 2018].
- [31] Apache Software Foundation, "Apache Griffin," 2018, <https://griffin.incubator.apache.org> [retrieved: November 2018].
- [32] L. L. Pipino, Y. W. Lee, and R. Y. Wang, "Data Quality Assessment," *Computing Surveys (CSUR)*, vol. 45, no. 4, Apr 2002, pp. 211–218.
- [33] D. P. Ballou and H. L. Pazer, "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems," *Management Science*, vol. 31, no. 2, 1985, pp. 150–162.
- [34] "Standard for a Software Quality Metrics Methodology," Institute of Electrical and Electronics Engineers, IEEE 1061-1998, 1998.
- [35] Oxford University Press, "Definition of Gold Standard in English," Online, 2017, <http://www.oxforddictionaries.com/definition/american-english/gold-standard> [retrieved: November 2018].
- [36] L. Ehrlinger, "Data Quality Assessment on Schema-Level for Integrated Information Systems," Master's thesis, Johannes Kepler University Linz, 2016.
- [37] B. Werth, "Identifikation von Datenqualitätsproblemen in integrierten Informationssystemen [Identification of Data Quality Issues in Integrated Information Systems]," Master's thesis, Johannes Kepler University Linz, 2016.
- [38] T. Haegemans, M. Snoeck, and W. Lemahieu, "Towards a Precise Definition of Data Accuracy and a Justification for its Measure," in *Proceedings of the International Conference on Information Quality (MIT ICIQ 2016)*, 2016, pp. 16.1–16.13.
- [39] J. R. Logan, P. N. Gorman, and B. Middleton, "Measuring the Quality of Medical Records: A Method for Comparing Completeness and Correctness of Clinical Encounter Data," in *AMIA 2001, American Medical Informatics Association Annual Symposium, Washington, DC, USA, November 3-7, 2001*, 2001, pp. 408–4012.
- [40] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986.
- [41] G. Vossen, *Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme [Data Models, Database Languages, and Database Management Systems]*. Oldenbourg Verlag, 2008.
- [42] O. Herden, "Measuring Quality of Database Schema by Reviewing – Concept, Criteria and Tool," in *Proceedings of 5th International Workshop on Quantitative Approaches in Object-Oriented Software Engineering*, 2001, pp. 59–70.
- [43] P. Oliveira, F. Rodrigues, and P. R. Henriques, "A Formal Definition of Data Quality Problems," in *Proceedings of the 10th International Conference on Information Quality (MIT ICIQ 2005)*, 2005.
- [44] H. Hinrichs, "Datenqualitätsmanagement in Data Warehouse-Systemen [Data Quality Management in Data Warehouse Systems]," Ph.D. thesis, Universitt Oldenbourg, 2002.
- [45] M. C. M. Batista and A. C. Salgado, "Information Quality Measurement in Data Integration Schemas," in *Proceedings of the Fifth International Workshop on Quality in Databases, QDB 2007, at the VLDB 2007 Conference, Vienna, Austria*. ACM, September 2007, pp. 61–72.
- [46] F. Naumann, J.-C. Freytag, and U. Leser, "Completeness of Integrated Information Sources," *Information Systems*, vol. 29, no. 7, Sep. 2004, pp. 583–615.
- [47] D. Ballou, R. Wang, H. Pazer, and G. K. Tayi, "Modeling Information Manufacturing Systems to Determine Information Product Quality," *Management Science*, vol. 44, no. 4, 1998, pp. 462–484.
- [48] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios, "Duplicate Record Detection: A Survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 1, 2007, pp. 1–16.
- [49] I. P. Fellegi and A. B. Sunter, "A Theory for Record Linkage," *Journal of the American Statistical Association*, vol. 64, no. 328, 1969, pp. 1183–1210.
- [50] J. Euzenat and P. Shvaiko, *Ontology Matching*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2007.
- [51] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys (CSUR)*, vol. 31, no. 3, 2000, pp. 264–323.
- [52] S. Melnik, H. Garcia-Molina, and E. Rahm, "Similarity Flooding: A Versatile Graph Matching Algorithm and its Application to Schema Matching," in *Proceedings of the 18th International Conference on Data Engineering*, ser. ICDE '02. Washington, DC, USA: IEEE Computer Society, 2002, pp. 117–128.
- [53] E. F. Codd, "A Relational Model of Data for Large Shared Data Banks," *Communications of the ACM*, vol. 13, no. 6, 1970, pp. 377–387.
- [54] J. Liu, J. Li, C. Liu, and Y. Chen, "Discover Dependencies from Data – A Review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 2, 2012, pp. 251–264.
- [55] C. Fellbaum, "WordNet and Wordnets," in *Encyclopedia of Language and Linguistics*, A. Barber, Ed. Elsevier, pp. 2–665, <https://wordnet.princeton.edu> [retrieved: November 2018].
- [56] DBpedia Association, "DBpedia," 2018, <http://wiki.dbpedia.org> [retrieved: November 2018].
- [57] S. Hoberman, *Data Model Scorecard*. Technics Publications, LLC, 2015.
- [58] S. Sadiq, T. Dasu, X. L. Dong, J. Freire, I. F. Ilyas, S. Link, M. J. Miller, F. Naumann, X. Zhou, and D. Srivastava, "Data Quality: The Role of Empiricism," *ACM SIGMOD Record*, vol. 46, no. 4, 2018, pp. 35–43.
- [59] Oracle Corporation, "Employees Sample Database," Online, <https://dev.mysql.com/doc/employee/en> [retrieved: November 2018].
- [60] Oracle Corporation, "Sakila Sample Database," Online, <https://dev.mysql.com/doc/sakila/en> [retrieved: November 2018].
- [61] Microsoft Inc., "Northwind and pubs Sample Databases for SQL Server 2000," 2018, <https://www.microsoft.com/en-us/download/details.aspx?id=23654> [retrieved: November 2018].
- [62] dofactory, "SQL Tutorial Sample Database," 2018, <https://www.dofactory.com/sql/sample-database> [retrieved: November 2018].
- [63] F. Naumann, "CD Datasets," 2018, <https://hpi.de/naumann/projects/repeatability/datasets/cd-datasets.html> [retrieved: November 2018].
- [64] Alpha Vantage Inc., "ALPHA VANTAGE," 2018, <https://www.alphavantage.co> [retrieved: November 2018].
- [65] L. Ehrlinger, "Data Quality Assessment for Heterogenous Information Systems," Online, <http://dqm.faw.jku.at> [retrieved: November 2018].
- [66] W3C Working Group, "RDF 1.1 Turtle," Online, 2014, <https://www.w3.org/TR/turtle> [retrieved: November 2018].
- [67] Oracle Corporation, "MySQL Connectors," Online, <https://www.mysql.com/products/connector> [retrieved: November 2018].
- [68] The Apache Software Foundation, "Apache Jena," Online, <https://jena.apache.org> [retrieved: November 2018].
- [69] DataStax, "Datastax Java Driver for Apache Cassandra," 2018, <https://docs.datastax.com/en/developer/java-driver/3.0/> [retrieved: November 2018].
- [70] Wikipedia, "Municipal District (Ireland)," 2018, [https://en.wikipedia.org/wiki/Municipal_district_\(Ireland\)](https://en.wikipedia.org/wiki/Municipal_district_(Ireland)) [retrieved: November 2018].

Protecting Against Reflected Cross-Site Scripting Attacks

Pål Ellingsen and Andreas Svardal Vikne

Department of Computing, Mathematics and Physics

Western Norway University of Applied Sciences

Bergen, Norway

Email: pal.ellingsen@hvl.no, andreas.vikne@student.hvl.no

Abstract—One of the most dominant threats against web applications is the class of script injection attacks, also called cross-site scripting. This class of attacks affects the client-side of a web application, and is a critical vulnerability that is difficult to both detect and remediate for websites, often leading to insufficient server-side protection, which is why the end-users need an extra layer of protection at the client-side, utilizing the defense in depth strategy. This paper discusses a client-side filter for Mozilla Firefox that protects against Reflected cross-site scripting attacks, while maintaining high performance. By conducting tests on the implemented solution, the conclusion is that the filter does provide more protection than the original Firefox version, at the same time achieving high performance, which with only some further improvements would become an effective option for end-users of web applications to protect themselves against Reflected cross-site scripting attacks.

Keywords—Cross-site scripting protection; input filtering; software security; injection attacks.

I. INTRODUCTION

A. Background

Cross-Site Scripting (XSS) has for long been among the top threats to Internet security as defined in numerous reports containing detailed information about the prevalence and danger regarding this class of vulnerability. Based on these existing results, a filtering solution for Firefox was first proposed by the authors of this paper in "Client-Side XSS Filtering in Firefox" [1]. This paper builds on the same work and expands on the results given there.

One of the reports that underpins the need for better XSS attack protection is the "Open Web Application Security Project (OWASP) Top 10 - 2017" report, which contains a list of the 10 most critical web application security risks [2]. Even though XSS has fallen to a 7th place in the "OWASP Top 10 - 2017" report [2], XSS still remains one of the most serious attack forms. Another report, being published annually for the past 12 years, by WhiteHat Security, called "2017 - WhiteHat Security Application Security Statistics Report" [3], also identifies that XSS is among the top two most critical web vulnerabilities. An interesting and troubling observation

made in this report is that even though XSS is considered one of the most critical vulnerabilities, it is not being prioritized for remediation by most websites. The statistics referred above suggest that the vulnerabilities receiving the most attention are vulnerabilities that are easy to fix, which is not the case for XSS. As a result of this, we would suggest that organizations must adopt a risk-based remediation process, which means that the most critical vulnerabilities should be prioritized first, like XSS. A report [4] published by Bugcrowd Inc., a web-based platform that uses crowd-sourced security for companies to identify vulnerabilities in their web applications, has analyzed the data from their platform, including information about the most common vulnerabilities found. The data in their report is based on all BugCrowd's collected data from January 2013 through March 2017, which contains over 96 000 submissions, where the by far most reported vulnerability is XSS with a submission rate of 25%. They also have data on the most critical vulnerabilities sorted by type, where XSS is considered the second most critical, which correspond to the same result found in WhiteHat Security's report. These are some of the most recent numbers regarding XSS statistics, but there have been published numerous studies on XSS vulnerabilities, attacks and its prevalence. One study by Hydara et al. [5] from 2014 conducted a systematic literature review where they reviewed a total of 115 studies related to XSS. They concluded that XSS still remains a big problem for web applications, despite all the proposed research and solutions being provided so far. As seen from the more recent numbers from OWASP, WhiteHat, and BugCrowd, this conclusion still holds true, that XSS vulnerabilities remain to be at large.

B. Problem Description

XSS vulnerabilities are caused by insufficient validation/sanitation of user-submitted data that is used and returned by the website in the response, which could compromise the user of the site. An attacker could potentially use this vulnerability to steal users' sensitive information, hijack user sessions or rewrite whole website contents displaying fake login forms. With the observation about how prevalent this type of attack is, and according to the mentioned WhiteSecurity

report that it is being not prioritized nor easy for websites to fix and remediate, it becomes clear that the user needs some means of protecting themselves at the client-side, since it is mainly the end-users of vulnerable web applications that are affected by potential attacks. Amongst the top 5 most used web browsers [6], Mozilla Firefox is the only browser, which does not include any kind of built-in filtering against XSS, which may compromise users in the case of a vulnerable web application.

In this paper we address this problem by creating a built-in filter protecting against Reflected Cross-Site Scripting (XSS) vulnerabilities inside the Mozilla Firefox browser. The choice of protecting against XSS for Mozilla Firefox is made for several reasons, one being that XSS vulnerabilities are among the most critical and prevalent web vulnerabilities in existence today with lacking protection mechanisms on both the server- and client-side of web applications [5] [3] [2]. This, in combination with the fact that Mozilla Firefox, which is the second most used web browser [7], does not provide a built-in filter for XSS protection, in contrast with the other major web browsers, Chrome, Edge, Safari and Internet Explorer, which do have such a filter built-in. The work of this paper will, therefore, be to create this filter built into and integrated with the existing source code of Mozilla Firefox, which is possible due to the fact that Mozilla Firefox is fully open source, allowing full access to the source code of the browser. This would be a case-study/pilot-case for the effect of building, integrating and running a filter protecting against XSS inside of Mozilla Firefox. As this is the second most used browser, with a market share of approximately 11.7%, as of the statistics from StatCounter's desktop browser market share worldwide for April 2018 [7], and the fact that XSS vulnerabilities are as prevalent as they are, it would be beneficial to look at a possible solution for adding this extra layer, the added filter, to the defense in depth strategy combining several XSS protection mechanisms for optimal overall protection.

For the work to be considered a possible usable solution, it needs to be evaluated thoroughly. There exists several different web browsers, all competing to being the best one, in terms of different factors such as performance, security, usability, customization and general look and feel. In such a competitive industry, web browser need to make sure that every included functionality is integrated and running as smoothly and efficient as possible, meaning an additional feature need to be well defined and robustly integrated. In the case of creating a filter for XSS, it needs to be secure, providing the necessary protection, and at the same time be efficiently integrated so the overall performance of the browser is not affected in any huge negatively direction. This means that the work needs to be evaluated in terms of at least two different categories, how well it protects against XSS attacks and how much it affects the performance compared to Firefox without the filter implemented. The overall validation of the filter

would be a qualitative research, as of how well the filter is implemented into the existing solution, but at the same time contain a quantitative method for measuring the performance of the filter, which could be accurately measured and compared to the original browser. By analyzing the performance number, however, it is not possible to correctly classify it as either right or wrong, but rather an estimation and analysis about if the added feature is in fact within reasonable limits to be considered as a well-performing solution.

C. Paper Outline

Section II will go into detail about web security and more specifically about XSS, explaining everything from what it is to different ways of protecting against it, focusing mainly at the client-side of web applications. This section will also include information about the current state regarding XSS prevalence and existing work, before ending with a detailed description of the methodology used in this paper. Following, in Section III, will be describing the web browser, Mozilla Firefox, which is the application that this paper is building on. Section IV-D7 will then describe all the design choices and the actual implementation of the work done, before Section V will contain an analysis of how well the work is done, in terms of protection effectiveness, performance and integration into Firefox. The Sections VI and VII, contains a conclusion based on all the work done, before ending with some suggestions for further improvement.

II. THEORETICAL BACKGROUND

A. Web Security

Web applications need to be protected against malicious users who want to steal and tamper their data. Web security is a broad concept, including many different aspects, protection mechanisms and potential outcomes. To be able to protect a web application, a basic understanding of information security is therefore needed, as it regards some basic principles and objectives for why security is important and how to utilize it correctly. Information security defines three important objectives of security [8], which are maintaining confidentiality, integrity, and availability. Confidentiality is about protection of information and data from being accessed by unauthorized parties. When someone gets access to data that they should not have access to, like sensitive information about users, it is considered a breach of confidentiality. Integrity is about the authenticity of information, ensuring that it is not altered and to make sure the source of the information is genuine. In web applications, this could be if an attacker is redirecting you to a different site than you originally intended to visit, as the site you get redirected to is not genuine. And lastly, availability regards that information should be accessible for the authorized users, which of course should be done in such a way that there is no breach of confidentiality or that someone might alter

the available data when accessing it. All these objectives of security are important when creating secure web applications. To be able to fulfill them all, web applications need to protect against several different attacks from malicious parties trying to steal their and their user's data. This is not an easy task, as there exist so many different types of attacks for targeting all kinds of vulnerabilities that are often contained in web applications.

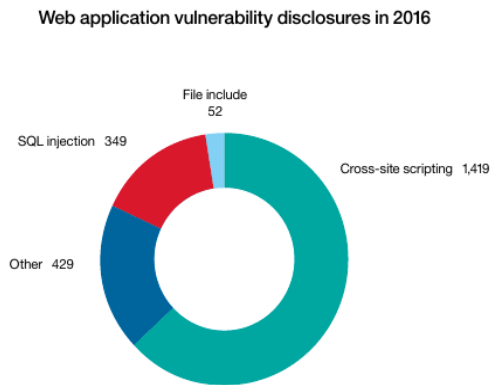


Figure 1: Web application vulnerability disclosures in 2016. Figure is taken from "IBM X-Force Threat Intelligence Index 2017" [9]

Several companies and organizations are doing annual research and assessment work containing a lot of collected data from a huge number of web applications and reports regarding security breaches and vulnerabilities. One of these reports [3], already mentioned in the introduction, from WhiteHat Security, goes into depth describing the current web security state. This report does not only contain information about XSS attacks, but a whole range of other web vulnerabilities with information of how prevalent they are, as well as which industries and areas that are the most vulnerable to different attacks. WhiteHat's report also contains a list of a vast number of web application vulnerability classes, describing 64 different web vulnerabilities that need to be protected against for web applications. This is a huge number of vulnerabilities, and while not all are relevant for every web application, many of them are critical, which needs to be addressed accordingly, where the injection attacks XSS and Structured Query Language injection (SQLi) is considered the most critical. Another report, by IBM, "IBM X-Force Threat Intelligence Index 2017" [9], is another comprehensive report containing statistics from different security events including web security, identifying what vulnerabilities are used and targeted industries. IBM also concludes that XSS and SQLi vulnerabilities are the most critical and prevalent, as seen in Figure 1, which need more attention from the different industries. As a whole, containing all web vulnerabilities, both reports have identified a small decrease in vulnerabilities in web applications, but also that attackers are targeting the most critical vulnerabilities more, in

which one of the most critical, XSS vulnerability is the least prioritized by applications to fix. Another concerning factor identified by both reports is that it takes too long to fix web vulnerabilities, which means both the application itself, as well as the end-users, are at a higher risk of being affected by a security breach.

The reports from WhiteHat Security and IBM, as discussed above, make it clear that the most prevalent attack on web applications is injection attacks, which includes attackers trying to break the confidentiality by stealing data from the web application itself or from the users of the web application. Injection attacks are performed with attackers inputting untrusted input to web applications that is executed as a command or query in such a way that it alters the course of execution, which could result in stealing of sensitive information or altering of data. There exist several types of injection attacks, but the most prevalent is by far SQLi and XSS. SQLi involves unauthorized users to inject Structured Query Language (SQL) commands that can read or modify data from a database connected to the web application. This is achieved through the usage of user-supplied input that gets used as part of a SQL query without the web application validating or encoding the input correctly. As attackers can read and modify data upon a successfully executed SQLi attack, it is possible to steal sensitive user data such as usernames and passwords, alter the contents of the stored information or simply delete everything contained in a database, which would incur huge complications for the affected web applications. The other critical vulnerability, XSS, will be covered in more depth in the following section.

B. Cross-Site Scripting (XSS)

Cross-site scripting vulnerabilities are caused by insufficient validation/sanitization of user-submitted data in the form of JavaScript code, that is used and returned by the website in the response without making sure the content is safe to use, which could compromise the users of the site. An attacker could potentially use this vulnerability to rewrite the contents on the website creating fake login forms to steal users' sensitive information, hijack user sessions or redirect them to other malicious websites.

There are three main types of cross-site scripting attacks, but there also exists some other defined types:

- Stored XSS, also called Persistent XSS
- Reflected XSS, also called Non-Persistent XSS
- Document Object Model (DOM) Based XSS
- Others - Plug-in XSS, Universal XSS, Self XSS

1) *Stored/Persistent XSS*: Stored XSS occurs when the injected script is stored on a publicly accessible area of a website, which means on the actual website itself. Typical places susceptible to Stored XSS attacks are in comment

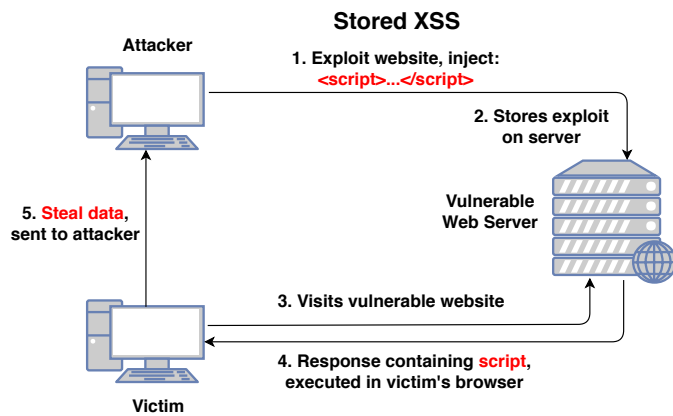


Figure 2: Stored/Persistent XSS

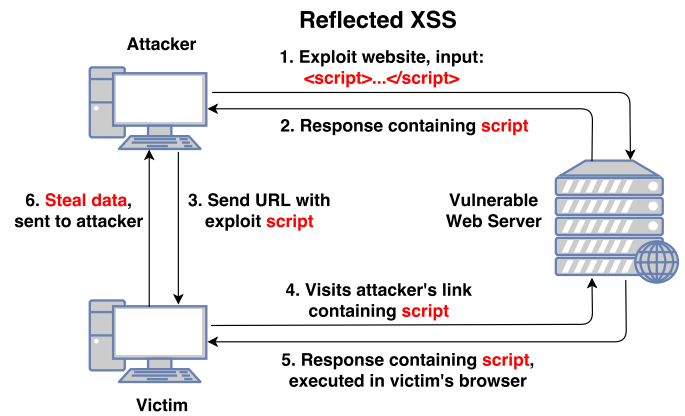


Figure 3: Reflected/Non-Persistent XSS

sections, message board posts or in chat rooms. Since the input data is stored in these places if the input data is an injected script, the injected script might get executed upon loading of the page, if the page is vulnerable. When a user visits one of these places, the browser will retrieve the data and render it, which in turn will execute the Stored XSS attack in the browser's context. Figure 2 illustrates the flow of a typical Stored XSS attack. Other places susceptible to Stored XSS attacks might include areas of a website only accessible to administrators, like a visitor log or other logs containing information about the usage of the website from users, as it is possible to inject JavaScript code into Hypertext Transfer Protocol (HTTP) headers [10] like the `Referer` [11] or `User-Agent` [12] headers. As the data from these headers are not unlikely to show up in some kinds of logs, a successful XSS attack here would be performed in the context of an administrator's browser, where it might be possible to not only get access to sensitive information from a single victim, but rather data from the whole web application. This type of XSS is very difficult to protect against on the client-side, as the client has no means to identify whether the JavaScript code coming from a website is legitimate, or if it is malicious JavaScript code injected by an attacker. A user does not need to visit any specific Uniform Resource Locator (URL) or include anything in the request to a website for a Stored XSS attack to be executed. From the client's perspective, all JavaScript code coming from a website is legitimate and should be rendered accordingly.

2) Reflected/Non-Persistent XSS: Reflected XSS occurs when the user input data is sent in a request to a website, which immediately returns data in the response to the browser, without the website first making sure the data is safe. Reflected XSS attacks are performed by entering data into search fields, creating an error message or by other means where the response use data from the request. In a Reflected XSS attack, the JavaScript attack code is not stored on the website itself, like it is in a Stored XSS attack. For a Reflected XSS attack

to work, the attacker needs to somehow make the victim request a special query, containing the malicious script. As mentioned, the search field is a typical input field that can be attacked. When searching for a query, the website often returns a page containing some results, which also will generate a unique URL containing the submitted query. This is how an attacker would create a specially crafted URL containing the exploit code, which then needs to be shared with a victim. If a user visits this particular URL, the attack code will run and execute in the user's browser. Figure 3 illustrates the flow of a typical Reflected XSS attack. As seen from this figure, a Reflected XSS attack contains a request to and response from a website, where the code inserted in the request is being used in the response. It is this particular data flow that protection mechanisms can take advantage of, where it is possible to compare the contents of the request with the contents of the response, to identify a potential attack. In this paper, this technique is utilized, which means it focuses on primarily stopping Reflected XSS attacks.

3) DOM Based XSS: DOM Based XSS is a type of XSS attack that in contrast to the other two types of XSS attacks only rely on JavaScript vulnerabilities on the client-side of the website and not the server-side. DOM Based XSS attacks exploits how a website uses JavaScript to dynamically change the DOM of a web page. The DOM of a web page is the structure of the page, containing information for the browser on how to render the page, with the usage of different HTML tags and attributes. The DOM of a page makes it possible for JavaScript code to interact with the page, making the page more dynamic. This also makes it possible for malicious code to change the page if JavaScript input is not handled correctly. If a website includes some JavaScript code in the response that directly uses input from an input source, like the URL, a DOM Based XSS might be executed. Figure 4 illustrates the data flow of a typical DOM Based XSS attack. These attacks can actually be performed without even sending the attack script to the web server at all, by using a special HyperText Markup

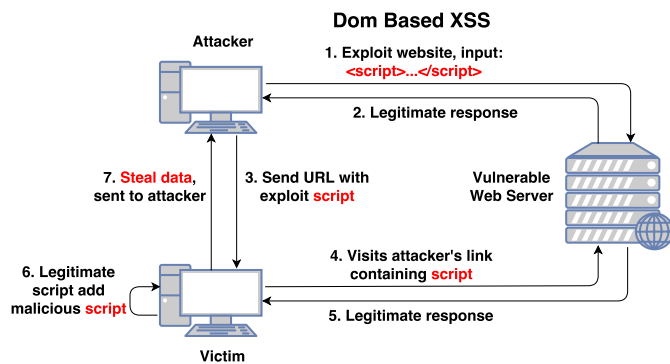


Figure 4: DOM Based XSS

Language (HTML) character, the fragment identifier #, in the URL. When using the fragment identifier, everything behind it will not be part of the request. This means that from the user inputs some data, to the malicious code is executed in the browser, the malicious code is neither part of the request nor the response of the website, but rather part of the DOM of the web page, if the content after the fragment identifier is used by client-side code in the response. DOM Based XSS is the least common type of XSS attacks, but it is also the most difficult to find and protect against. Since the attack only relies on flaws on the client side, by using JavaScript code, server-side filtering cannot detect this attack at all, which is a good reason why it is necessary to have protection also on the client-side of a web application.

4) Other XSS Types: Although there exist three main types of cross-site scripting, as these attacks have evolved and been used in different ways, XSS types could now be categorized into some additional sub-categories, Universal XSS, Plug-in XSS and Self XSS.

a) Universal XSS: Universal XSS [13] is a form of XSS attack that exploits the browser itself, browser extensions or website extensions in order to exploit a website. Universal XSS is a very dangerous type of XSS as it does not exploit the website directly, meaning that a website does not need to contain any vulnerabilities to be exploited. Modern web browsers support extending their functionality by utilizing plug-ins, small programs that add more features to the browsers. There also exists plug-ins that are not loaded through the browser but by the website itself. These plug-ins often have access to the contents of the websites, and often require input from the user for its functionality to work. By having user input in combination with features for displaying or editing contents on a web-page, the plug-in might create an opening for allowing a cross-site scripting attack against the web-page it is being used on. An example could be a plug-in that allows websites to display pdf-files. If an attacker injects some JavaScript code in the filename of the displayed pdf-file, this JavaScript code could be rendered in the browser, if the plug-in does not

have proper validation and encoding for the input field used for the filename. XSS vulnerabilities introduced by insecure plug-ins are often categorized as Plug-in XSS, which could be considered as a sub-type of Universal XSS.

b) Self XSS: Self XSS is when users themselves create and execute the attack in their own browsers, which can not exploit other users, as in the case of the three main types of XSS. Self XSS is mostly a social-engineering attack used to trick users into executing XSS attacks on themselves, often by making them copy and paste JavaScript code into their own browsers. Awareness around this particular attack was gained through the popular social media website Facebook.com, as this attack became quite widespread against the users of their site, which led to Facebook publishing a warning [14] against Self XSS scams. Facebook even created a warning displaying when a user opens the developer console window in their browser, while visiting their site facebook.com, to mitigate the attack.

As described above, XSS attacks occur because web applications are using unsanitized input data when displaying and rendering content. For a successful XSS injection, from the attacker's perspective, the input containing the malicious JavaScript content needs to be entered into the web application in a way that it is somehow gets executed in the browser. The next sections will explain how this is done, and give some examples of how typical XSS attacks are performed.

When performing an XSS attack, it is possible to inject the malicious script into the web application by using several different input sources. An input source is considered an entry point for user input to enter into the application. The most common input sources for XSS attacks are from the GET- and POST- parameters, which most often comes from HTML input elements. A typical example is the search field found on many websites, which most often is an HTML input tag. After using the search field, the search query is likely to be included in the URL of the returned web page, which would consist of a GET parameter containing the query. HTTP headers is another input source for script injections, as discussed in Section II-B1. Injecting script content through HTTP cookies, which is a small piece of data sent to the user's web browser from a server, is also an option, although this is much less common, as a potential attacker would most likely need to get access to other users' cookies for injecting their script. Since the end goal of an XSS attack often includes getting access to such cookies, using them as an input source for an attack seems less likely, although in theory, it is still a possibility.

For a successful XSS attack, the injected script content needs to be entered into the web application in a way that would actually render the script in the browser. This could be done by using a wide variety of attack vectors, depending on how the web application uses the input when generating the response. Attack vectors are typically a combination of

HTML tags that include the script to be injected and executed. These tags could either embed the script content directly or reference an external resource containing the JavaScript code. The most common attack vector is the usage of the `script` tag. Another very common attack vector is the usage of the `img` tag in combination with `on-event` handlers [15]. The `on-event` handlers are properties that let HTML elements react to events, where events are different actions like when an element is being clicked, getting focus, or when it is loaded. The reaction to an event can be specified to load script content, which is why they are often used in XSS attacks. OWASP's "XSS Filter Evasion Cheat Sheet" [16] is a comprehensive list of attack vectors utilizing a lot of different techniques, including many uses of `on-event` handlers. Other than the most common `script` and `img` tag, the `iframe`-, `body`-, `svg`-, `object`- and `style`- tag are also HTML tags not uncommonly used in XSS attacks. OWASP's list [16] contains descriptions of these and many more, including techniques to hide the injected script from being detected by potential XSS filters.

c) Example attack: A typical scenario for an XSS attack starts with an attacker looking for input fields on a web page where the submitted data is output without being encoded. As mentioned above, the search field is a common input source. An attacker could, therefore, exploit a vulnerable search field, with the intention of trying to hijack another user's session. The search field is often exposed for an attack, as when you input a query, the same query is most likely being returned and rendered by the website. If this input is not properly being encoded, it could allow the attacker to input JavaScript code that is being executed in the browser's context when the website returns the query, which could be achieved using the `script` tag as the attack vector. For hijacking a user's session, the attacker would need some JavaScript code that extracts the user's session data, typically found in a cookie from the logged in targeted user. The exploit code, `<script>document.location='http://attacker/cookieStealer.js?c=document.cookie</script>`, could then be inserted into the search field. After creating this exploit, the attacker would need to copy the URL from the result page after doing the search. Since this is a Reflected XSS attack, the attacker would then need to share this URL to potential users of this exploited site. If a targeted logged in user now visits this particular URL, the user's session cookie is being sent to the attacker. The attacker could then use this cookie to log in onto the exploited website, which means the attacker would be impersonating the user.

Another popular XSS attack is to rewrite the contents of a website, creating fake forms for tricking users to enter sensitive data like credit card information or login details. The attacker would then make these forms submit the sensitive data to themselves, rather than to the exploited website.

A typical thing that XSS attacks have in common is that they are often not easy to detect by the end-users themselves. In case of both the cookie stealing and fake forms exploits, the attacker could simulate the actual behavior of the exploited website, making it almost impossible for users to detect that they have been compromised. By having a client-side filter in the browser, a user could not only be notified of a potential attack, but the filter could also completely stop it from occurring in the first place, which is the intent of the filter.

C. Counter-Measures

There exist many counter-measures for XSS attacks, consisting of several techniques as well as more specific policies to follow, for securing web applications. It is highly recommended to utilize a variety of many different counter-measures, as it might be challenging to implement them being completely robust and secure from unknown attacks and not all policies are fully supported by all web browsers.

a) Validation/sanitization: The first step towards protecting against XSS attacks is to make sure that valid malicious code does not enter the web application at all. Validation/sanitization of all untrusted data input to a web application makes sure that malicious input is either being rejected or manipulated into being safe for usage in the response from the website, used in the output of users' browsers. It might be difficult to implement this properly as it can be challenging to know what a malicious input looks like, considering all the possible attack vectors that use advanced obscuration techniques. A common mistake is to rely only on blacklist validation, which is often trivial for attackers to circumvent, by utilizing alternative input variations. White-listing is in general considered much safer, only allowing the characters that the web application should accept, for example, an integer or a date. In case of free-form text input, white-listing becomes difficult, as the users should be allowed to enter almost any character, hence the free-form. Any validation technique becomes ineffective and difficult to implement in the case of free-form text, which is why input validation should not be used as the primary defense against cross-site scripting attacks, and why output encoding is needed.

b) Output encoding: Output encoding is the most effective remediation for cross-site scripting attacks when done properly. Output encoding should be implemented every place untrusted input is being outputted and rendered in the browser, making sure the input is displayed as data and not executed as code in the browser. It is important to implement the output encoding according to the context it is being used in, because different encodings are needed depending on the context used. JavaScript, HTML, and URL's all use various encodings, which is why there is no single solution to how output encoding should be implemented. Typical strategies are

to escape unicode, a typical character encoding, converting unwanted characters to benign equivalents, percent encoding and escaping hex values, as described in more detail in OWASP's XSS (Cross Site Scripting) Prevention Cheat Sheet [17].

c) Content Security Policy (CSP): Another powerful counter-measure is Content Security Policy (CSP), which is a declarative policy that let web application owners create rules for what sources the client is expecting the application to load resources from. To enable CSP, the web server needs to utilize the `Content-Security-Policy` HTTP response header [18], where the policy for the application is specified, including desired directives. Each directive describes a policy for a certain resource type or policy area, for example to prevent inline scripts from running, only allowing content to be loaded for some trusted domains or restricting all content to only load from the site's own origin. CSP also have a reporting feature, which means when a policy is being violated, it is possible to get a report sent to the desired location, containing information about the violation. This could be helpful for web application owners to know if their policies are too strict or needs modifications, as a policy can consist of many different directives. Even though CSP can stop most cross-site scripting attacks by utilizing a set of well-defined directives, it is stated in the World Wide Web Consortium (W3C) Recommendation [19] that CSP is not meant as a first line of defense mechanism, but rather an element in a defense in-depth strategy, as an added layer of security. A study by Weichselbaum et al. [20] was done in 2016, including 1,680,867 hosts with 26,011 unique CSP policies, observing that 94.68% of all policies that attempts to limit script execution are ineffective, as well as 99.34% of the hosts have policies that offer no benefit against XSS at all. This is a very clear indication that CSP in practice is difficult to utilize correctly and this is why it should not be used as the primary defense against cross-site scripting attacks.

d) Same-origin policy: Same-origin policy [21] is a policy implemented inside web browsers that isolates potentially malicious documents by restricting how a document or script loaded from a specific origin can interact with resources from other origins. For two web pages to have the same origin, they need to have the same protocol, port and host, which means they are allowed to load resources from each other. Cross-site scripting attacks often involve the usage of different external JavaScript files for collecting data from compromised users, which could be blocked by utilizing the same-origin policy.

e) HTTPOnly cookie flag: As mentioned in Section II-B4c, cookies could contain valuable information for attackers, which means they should be protected from unauthorized access. The `HTTPOnly` cookie flag is an additional flag included in the `Set-Cookie` HTTP response header [22], preventing JavaScript code from accessing the contents of cookies. This is not considered a counter-measure for XSS,

but rather for mitigating the risk of an attacker accessing other users cookies in the case of an attack.

f) Disabling JavaScript: A more drastic approach that would effectively stop XSS is to disable JavaScript, since these attacks rely on a JavaScript environment for execution. This solution can be effective for simple static websites, but most dynamic websites require some sort of JavaScript support for basic functionality, which means this remediation would not be suited as a general solution.

D. Cross-Site Scripting Filters

Filters try to stop cross-site scripting attacks by utilizing a set of rules to detect potentially malicious input data, before either blocking it or sanitizing it for safe usage. There exists many XSS filter implementations, with varying focus on the different areas such as security, performance, low false-positives and usability. All of these areas are in focus in most filters, but it is not common for a filter to be best in all categories, as they do not necessarily compensate each other. There is, however, one clear way to differentiate between filters, which is to divide them into two groups, server-side and client-side filters:

a) Server-side filters: Server-side filters are implemented on the server side of a website, which means it can only detect input data that are sent via the server. The DOM Based XSS attack is possible to perform without sending the attack code to the server at all, as discussed in Section II-B3. This means a server-side filter would not be able to detect the attack at all, which implies it would not be able to stop the attack. There are several existing server-side filters, which typically needs to be integrated into the source code of the web application. A study made by S. Gupta and B.B. Gupta [23] has a quantitative discussion for server-side filters, discussing some of the state-of-the-art techniques they are using. The study concludes that there are generally several flaws with server-side filters that need to be addressed, like too much altering of existing code-base, long learning phase, as well as too many false-positives and false-negatives. The study also emphasizes that server-side filters do not detect DOM-based XSS attacks. With all the combined flaws and design limitations of server-side filters, it becomes evident that only relying on server-side protection is not enough, and why it is necessary with client-side filters as an extra layer of security.

b) Client-side filters: Client-side filters are located in the client, which typically would be the web browser used to access web applications. Client-side filtering could be able to detect DOM Based XSS attacks, providing the extra protection server-side filters are missing. However, even though client-side filters could possibly detect all types of XSS attacks, they should not be used without server-side filters. By placing the filter on the client-side, it means that the user might be able to modify it to circumvent the filtering. It is, therefore,

strongly recommended to utilize both server- and client-side filtering, to be able to protect against all attack types of XSS and achieving good protection following the defense in depth strategy. This paper focuses on client-side filtering, which includes a discussion of various existing solutions, presented in the next sections.

Regular Expression Based Filters Using regular expressions is a popular technique for client-side filters, where the filter is typically located between the network layer and HTML parser in the browser. Regular expressions are then used to identify potentially malicious code in the HTTP requests and to approximate the rules of the HTML parser to know which content in the HTTP response that would be treated as script content [24]. By doing these approximations, the filter do not have to recreate the browser's own HTML parser, which would lead to the HTTP response being parsed twice, first for the filter to identify and remove potential malicious code and then for the browser to parse the page as normal. These approximations do, however, have their drawbacks, as they incur a higher number of false positives, due to several flaws in their design [24]. These flaws are a consequence of attackers trying to make the content from the request, the actual attack code, differ from the response so that the approximation rules would not detect it as an attack. Some common flaws are that the filters do not correctly approximate the decoding process of different encodings or do not take into consideration that different characters can be used to delimit HTML attributes.

A popular client-side XSS filter using regular expressions is an extension called NoScript [25], for the Mozilla Firefox browser, first released in 2005 and actively updated by the maker Giorgio Maone. The filter is matching HTML code for injected JavaScript in the request by utilizing regular expression rules for simulating the HTML parser, which would potentially lead to false-positives, as it is better to over-approximate these rules than to let an attack bypass the filter [24]. Due to a lot of false-positives, NoScript try to solve this by prompting the user to repeat the request with the filter disabled, allowing the user to decide for themselves if they think it was a false positive. This is a decent approach for security-aware users, but in general, users do not have the knowledge or desire to take action in the case of security-related issues [26]. **String-matching Based Filters** String matching is another method for client-side XSS filtering, used by the filter in the Google Chrome browser, called XSS Auditor. XSS Auditor works by matching the HTML code for injected JavaScript code from the request with the response from the website, after it is been parsed by the browser's own HTML parser [24]. This means that XSS Auditor does not need to approximate any of the HTML parser rules, since the parsing is already done when the matching algorithm starts. This is achieved by the location of XSS Auditor, which is between the HTML parser and the JavaScript engine, as shown in Figure 5. This placement makes it possible to block scripts after parsing, by blocking them from

being sent to the JavaScript engine for execution. The location of XSS Auditor have benefits like performance, by not having to simulate the HTML parser, and the fact that the JavaScript engine has a narrow interface it is reasonable to assure that all scripts are being processed by the filter before being executed.

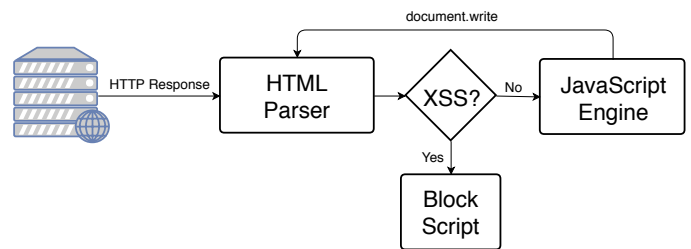


Figure 5: XSS Auditor design

XSS Auditor also has some limitations, some of which are discussed in the paper from Stock et al. [26], which lists several flaws with the design and string-matching algorithm used in XSS Auditor. As mentioned in the paper, these are mainly flaws regarding protection of DOM-based XSS, which is not the main type of attack that XSS Auditor is protecting against. It is, however, relevant to take notice of these limitations, as it might be desirable to not make the same limitations when designing and implementing a new filter.

Scope issues are related to the fact that XSS Auditor does not support every type of XSS or are neglecting functionality that enables XSS attacks. One example being that XSS Auditor relies on encountering dangerous elements during the HTML parsing of the response, which is not always the case, for example, when a web page is using the JavaScript function `eval()` [27]. `eval()` is a function that evaluates the string representation of JavaScript code inserted inside its parentheses, which means if `eval()` uses data from the URL of the loaded web page, this evaluation could be done without entering the HTML parser, which means that XSS Auditor would not detect it.

Another flaw in XSS Auditor is that some special characters needs to be present in the request for the filter to be activated. If any of these characters are not present, the filter deactivates. As the paper describes, it is possible to successfully execute an XSS attack without any of these special characters being used at all.

Double injections is yet another limitation that XSS Auditor does not protect against, which is the inability to detect attacks containing concatenated values coming from more than one source of user input. An attacker could use two different input sources due to application-specific code that concatenates two or more user inputs. When creating an attack using double injections, the exploit code consist of two or more parts, but gets executed as one concatenated attack code. Since XSS Auditor's string-matching algorithm checks for the whole

script code, the algorithm would not detect the attack, as the whole script code does not exist from any single user input source.

1) *State of Current Browsers:* Regular expressions and string matching are among the techniques being implemented in the top five most used web browsers for desktop, which according to the desktop browser market share worldwide from StatCounter [7] are Chrome, Firefox, Internet Explorer/Edge and Safari. Table I contains information on the state of their XSS protection status. Both Chrome and Safari use the mentioned string matching based XSS Auditor filter. XSS Auditor was first built into the browser engine WebKit, which Safari uses, before also being integrated into a fork of WebKit called Blink, which Chrome uses. Internet Explorer and Edge both have a filter implemented based on the regular expression technique, first introduced in Internet Explorer 8 [28]. Firefox, however, being the second most used web browser, does not have a built-in filter, but rather relies solely on CSP support, which again relies on websites to properly define the CSP rules. By not having a client-side filter, the defense in depth strategy is also weakened, where a potential filter would provide an extra layer of security for the end-users of the application.

Table I: Top 5 Web Browsers XSS Protection Status.
Data retrieved from Mozilla [18]

XSS Protection					
Built-in filter	✓	✗	✓	✓	✓
CSP	✓	✓	✓*	✓	✓

* Limited support

III. MOZILLA FIREFOX

Mozilla Firefox is a free and open-source web browser developed by Mozilla, with its first major release in 2002 [29]. Firefox's source code has a layered architecture where the code is organized as separate modular components. Firefox is multi-threaded and follows the rules of object-oriented programming, where access to internal data is achieved through public interfaces of the classes [30]. One of the primary requirements of Firefox is that it must be completely cross-platform, which is why the browser consists of several components focusing on this area, like making sure the operating system dependent logic is hidden from the application logic.

This section will explain some of the most relevant parts of Firefox, with regards to the filter described in this paper. The parts explained have been slightly simplified, making it

easier to understand the relation of how everything is working together, again with regards to the added XSS filter.

A. Firefox Overview

The main components of Firefox can be divided into the user interface XML User Interface Language (XUL) and the browser and the rendering engine Gecko. XUL is Mozilla's own language for building portable user interfaces, which is an Extensible Markup Language (XML) language [31]. Gecko is Mozilla's browser engine built to support many different Internet standards, including HTML 5, Cascading Style Sheets (CSS) 3, DOM, XML, JavaScript, and others. Gecko contains many different components for document parsing (HTML and XML), layout engine, style system (CSS), JavaScript engine called SpiderMonkey, image library, networking, security, as well as other components [32].

Mozilla also has a build system [33] using the `make` tool [34], consuming `Makefiles`. The command-line interface `Mach` [35] is used to help developers perform common tasks for working with the Mozilla codebase, making it easy to start building, debugging and testing Mozilla projects.

Firefox consists of over 36 million lines of code [36], written in several languages, which are mostly C++ and JavaScript, but also HTML, C, Rust, XML, Python and Java, as well as other less used. The source code directory of Firefox [37] contains many folders where the code is grouped based on their functionality. Some of these groups consist of functionality related to document parsing, JavaScript execution, image loading, extensions, and networking, just to mention a few. Mozilla also has strict rules about *how* the code should be implemented, not just how it is structured into directories. As mentioned above, Firefox is object-oriented, using a lot of public interfaces. They have also implemented several utility- and helper-classes for writing specific functionality inside their code-base. Although the source code is mostly written in the C++ language, which provides this functionality built-in, Mozilla uses many of their own methods for these functions. This means that it is necessary to acquire specific knowledge regarding these coding rules before attempting to make changes to the Mozilla codebase, as it is a complex piece of software.

1) *Loading of a Web Page:* As mentioned above, Firefox consists of several components, including its rendering engine Gecko, which is the most relevant part for the implementation of this filter, as it contains everything related to document parsing and handling of JavaScript execution. Figure 6 is a simplified description of the loading of a document in Firefox, containing only the relevant parts which are important regarding the XSS filter. When a typical HTML web page is loaded through Firefox, two internal document classes, `nsDocument`, and `nsHTMLDocument`, are created, controlling the creation and representation of the web page

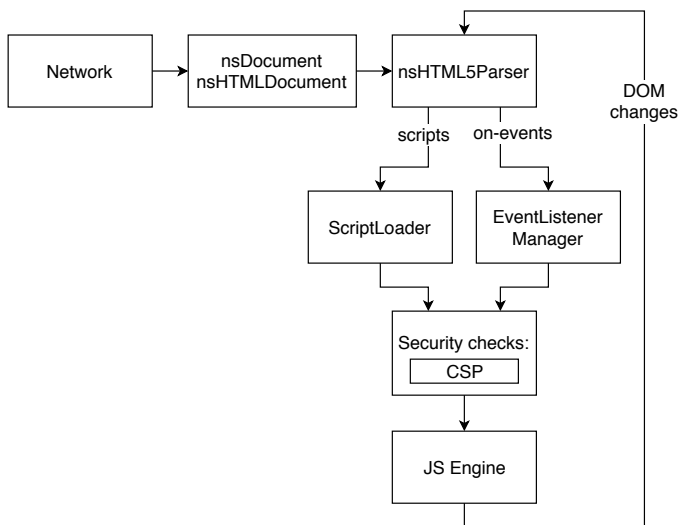


Figure 6: Simplified data flow for rendering a web page

to be loaded. These documents are responsible for creating and calling all the relevant parsers, like the HTML parser [38], `nsHtml5Parser`, as well as initializing the script executioner class, `ScriptLoader`, which is responsible for handling script content coming from `script` tags. The HTML parser receives data from the network that needs to be parsed. Every time the parser encounters some script content, the relevant parts of Firefox that handle this content is invoked. In the case of `on-event` handlers, the `EventListenerManager` class is invoked. A common source for script content is the `script` tag, where the script loader class, `ScriptLoader`, would be invoked with the discovered script. The script loader class will then try to extract the script and either execute it as an inline or external script. Before the script is passed to the JavaScript engine for execution, a security check is performed for finding out if the script is allowed to run. This security check involves checking with the CSP rules if it is allowed to load if these rules are specified by the loaded website. If the script passes this check, it will be handed over to the JavaScript engine which will execute the script in the browser. The HTML parser will continue parsing the data entering through the network, repeating the steps when new script content is discovered.

B. Security Mechanisms

Firefox includes many internal security mechanisms for making sure that the browser itself is not being compromised by attackers, as Gecko loads JavaScript content from untrusted and potentially malicious web pages, which then again run on the user's computer. These security mechanisms include several complicated concepts regarding same-origin policy, compartments, and principals, all explained in detail at Mozilla's own website [39]. This section will try to give a simplified explanation of why all these concepts are important

and how they are used. The reason why this is interesting to look at is because a countermeasure for XSS, CSP, is implemented inside Firefox using the principal concept. Since CSP provides similar functionality as the work described in this paper provides, the filter created should also ideally be implemented in a way that follows the same principles, fulfilling the necessary security requirements.

1) Same-Origin Policy: The same-origin policy is restricting how a document or script loaded from a specific origin can interact with resources from other origins, as described in Section II-C0d. The security model for web content is based on this policy, which is also used inside Firefox as a script security mechanism [39]. As Firefox's rendering engine Gecko consist of different languages, its core in C++ and its front-end in JavaScript, these two parts need to interact with each other in a secure manner. The JavaScript front-end is actually running with system privileges, meaning that if it is compromised, attackers might get control of the user's computer. As this JavaScript code is interacting with web content from web applications, which again is susceptible to XSS attacks, it is important to make sure that JavaScript code from Gecko itself is not affected by any such attack, which is achieved by utilizing the principle of the same-origin policy.

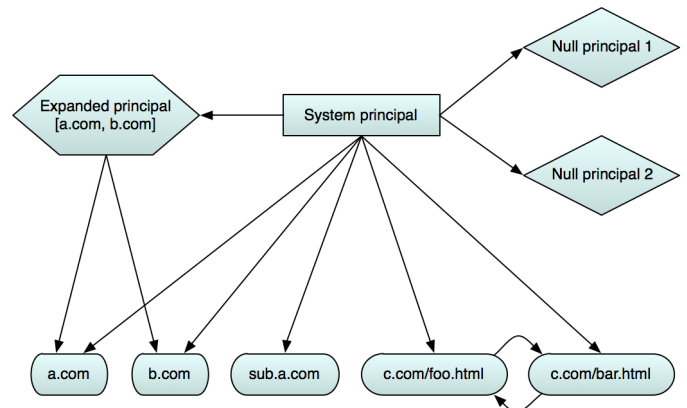


Figure 7: An overview of the relationships between the different security principals.

Figure 7 is taken from Mozilla's website, about "Script Security" [39]

2) Compartments and Principals: A security measure in Gecko is that it is divided into different compartments. Compartments could either be internal parts in Gecko or a content window, a typical website, where different parts can only access other parts if they are in the same compartment. The concept of compartments is, therefore, using the same-origin policy principle. Every part inside a compartment is, therefore, same-origin with the others and no additional security checks are performed when parts inside the same compartment talk to each other. If Firefox loaded the website at <http://example.com/subfolder/>, all the HTML

elements and script content residing on this exact address would be inside the same compartment. There are, however, different ways for compartments to access parts of other compartments, where the main rules are that higher privileged compartments have access to less privileged compartments, but not the opposite, unless the higher privileged compartment explicitly chooses to share its access.

To be able to determine the security relation between different compartments, a concept called *security principals* is used, which is something every compartment have. Figure 7 illustrates the relationship between different principals, as there are several different principals, each with its own rules. System principals pass all security checks, which is what the JavaScript code from Gecko is running with. Content principals are associated with web content, meaning that content from a specific origin could access parts from content inside the same origin. An expanded principal is specified as an array of origins, meaning that it contains several content principals. The expanded principal itself gets access to its contents, but the content principals within does not get access to the expanded principal. Finally, there is the null principal, which fails almost all security checks, meaning it has no privileges and can only be accessed by itself and the JavaScript code from within Gecko.

3) *Content Security Policy (CSP)*: Content Security Policy (CSP), as described in Section II-C0c, is a security feature that is also implemented in Firefox. Since CSP is part of the script security model, it also has a principal. This means that CSP is created through a principal and access to it needs to be done through a principal. The main class, the `nsDocument` class, is the place where the CSP is initialized, by using a principal. As the `nsDocument` creates and holds a reference to the CSP Principal, other classes can get access to the CSP through the `nsDocument` class. Some noteworthy places that CSP is used inside Firefox are the script loader class, `ScriptLoader`, and the `EventListenerManager` class. These are locations which handle content related to script execution, and therefore also the place where the proposed filter should be placed.

IV. DESIGN AND IMPLEMENTATION

This section will go through everything from the development process of the implemented filter, including the requirements, design, tools used and the actual implementation of the solution.

A. Design Choices

Software development includes a lot of choices that need to be made during the development lifecycle, regarding analyzing the problem, coming up with a solution, making the design and figuring out how it should be implemented. When creating a filter for Firefox defending against cross-site scripting attacks,

it is possible to choose many different approaches towards the same main goal, but yet achieving differently in different categories such as performance, availability, usability, maintenance and of course security. In this section, some of the design choices made for our solution will be explained in detail.

a) *Usability*: The filter should be easy to use, by not requiring any user-interaction at all. The NoScript plug-in for Firefox, mentioned in Section II-D0b, is an example of something that is not wanted, as NoScript do require a fair amount of user interaction, as the plug-in have a lot of false-positives. In a worst-case scenario, a user might accidentally allow an attack to get executed, even though the filter did stop the attack and warned about it, as users might not understand what it means and the risk of ignoring the warnings.

b) *Low false-positives*: It is important that the filter do not interfere with a user's normal browsing sessions, unless it is to protect the user from an actual attack. To achieve this, the filter should have a low number of false-positives, which means that the filter should minimize the number of times where it think there is an attack when in reality it is not. The opposite of a false-positive is a false-negative, which is when the filter thinks a script is safe to load when in reality it is an attack and should be blocked. In practice it is difficult to guarantee both non-existent false-positives and false-negatives in a filter meant for defending against cross-site scripting attacks, as there are so many different ways of using JavaScript in web applications, which again is one of the reasons why cross-site scripting attacks are so prevalent. There is, however, a balance to be made, to make sure that the filter do protect against most attacks, which means it might introduce some false-positives, but at the same time it cannot be too strict either. An example of a too strict filter is again the NoScript plug-in for Firefox, which is really aggressive, introducing a lot of false-positives which would interfere a lot during normal browsing sessions, again requiring user interactions as a workaround.

c) *High performance*: The filter should not incur a lot of performance overhead, which would make the loading of web pages slower, which again would interfere with the usage of normal web browsing. When using the filter, there should be no noticeable delay when loading web pages in comparison with the version of Firefox without the filter. This is an important requirement, because of the competition between web browsers, as discussed in Section II.

d) *Provide protection against Reflected XSS*: The whole point of a filter protecting against cross-site scripting attacks is to provide this protection properly. As there exist several different types of XSS, as discussed in Section II-B, it is important to clarify that the main focus of the filter is to protect against the Reflected XSS type. This is the type of XSS that filters for the other major web browsers also primarily focuses on, as it is very prevalent and the easiest to discover, as described in Section II-B2. It is, however, desirable to also

protect against DOM Based XSS, which there will be some basic protection against, as a byproduct of the Reflected XSS protection. Complete DOM Based XSS support will, however, be lacking, as in the case of XSS Auditor, as explained by Stock et al. [26].

1) Browser Extension vs Internal Implementation: The main goal of this work is to add some functionality to the Firefox browser, which there are several ways of accomplishing. Firefox do provide support for browser extensions [40], which can extend and modify the capabilities of the browser. These extensions are built using JavaScript, HTML, and CSS by using the WebExtensions API, a cross-platform system for developing extensions. They can provide a lot of functionality for altering the contents of or extracting information from a web page, either with or without required user interaction. There are, however, some reasons why browser extensions are not suitable for this, explained in the following paragraphs.

a) Availability: The main reason why browser extensions are less suitable is because they are something that users themselves need to find, install and use. It should not be necessary for users to know about what cross-site scripting is and why it is important to protect against it, for them to take advantage of this filter. By making this protection a choice for the user, the filter would most likely not be used by the majority of users. This is why an integration with Firefox itself would be a better solution, as then all users would take advantage of the filter without the need of any knowledge about it or action required.

b) Performance: Even if there are users choosing to install and use such a security filter, there is another drawback by making it as a browser extension, which is a performance issue. When creating a browser extension for Firefox you can only use the API's supported by Firefox [41], utilizing JavaScript code that talks to the internals of the browser itself. This means there are more layers that the data needs to go through, from getting from the filter to the internals of Firefox, which is needed for the functionality of the extension to work. If the filter, however, is placed inside the internals of Firefox, some redundancy will be removed, which again will lead to a better performance, which is what is chosen for this filter design.

c) Security: The purpose of the proposed filter is to protect against Reflected XSS attacks, which means the injected script is contained in both the request and response. By implementing the filter as a part of the internal implementation of Firefox, it is easier to have a more robust integration being more secure, as Firefox has a lot of coding principals including many security features, as described in Section II.

2) Blocking Technique: When detecting an XSS attack, the filter needs to take action to block the injected script. There are mainly two ways of doing this, either blocking only the injected script or blocking the whole web page from loading.

By only blocking the injected script you interfere less with the browsing experience of the user, as they can still use the website as normal, without the parts potentially affected by the injected script, which is what has been chosen for this proposed filter.

3) Filtering Technique: As discussed in Section II-D, there exists XSS filters based mainly on the two filtering techniques regular expressions and string matching. For this paper, the string matching technique and design from XSS Auditor was chosen as the main basis. XSS Auditor used in the Google Chrome browser does achieve high performance, few false-positives and low interference with normal web browsing, providing protection against mainly Reflected XSS attacks, as desired from the requirements in this paper.

B. Design Overview

The main design of the filter is to compare every script returned in the response with every potential dangerous script from the request. If there is an occurrence of a script appearing in both the request and response, the cross-site scripting filter will block this particular script from being executed. The filter itself is structured as its own class inside Firefox's source code, which makes it easy for other components in Firefox to use the filter when needed. The filter is placed after the HTML parser, but before script execution, providing benefits regarding both security and performance. The following sections will describe the design of the filter in more detail.

1) Placement: By basing the solution on the filtering principals of XSS Auditor, the placement in Firefox will also be similar to how Auditor is placed inside of Google's Chrome browser. Auditor is placed between the HTML parser and JavaScript execution environment, which provides several benefits, regarding high security and performance, as explained in Section II-D0b.

The filter needs to know what Firefox would intercept as script content, to be able to filter on the correct data. If the filter was placed before the HTML parser, the filter would need to simulate the rules of the parser to try to approximate and identify what Firefox would intercept as script content. This means that each loaded document would be parsed twice, once from the filter and once from Firefox's own parser, which would incur a lot of performance overhead. Since Firefox need to parse the HTML documents regardless of the filter's presence, by placing the filter after the HTML parser, it can use the results from Firefox's own parsing when determining which content to filter on, which again would not add any extra performance overhead regarding the actual parsing process. Since the filter does not need to approximate the parser rules when placed behind the HTML parser, the filter can also be sure that it will discover, identify and act upon all the scripts entered through Firefox, as the parser in Firefox will properly identify all script content before they are processed further. As

explained in Section III, script content from `script` tags and on-event handlers get sent to the classes `ScriptLoader` and `EventListenerManager`, which will further examine the data and conduct the necessary security checks before they are sent to the JavaScript engine for being executed, as shown in Figure 6. By extending on this figure, extracting the relevant parts, Figure 8 shows the placement of the XSS filter, residing in the same location as the CSP security feature.

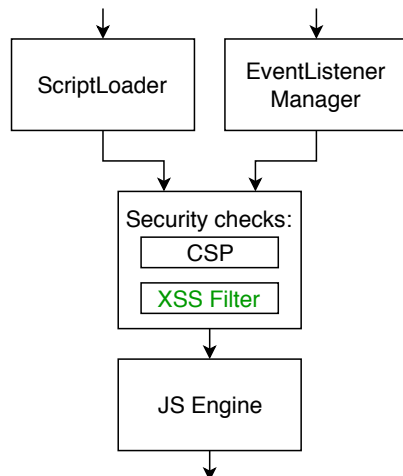


Figure 8: XSS Filter placement inside Firefox

2) *Filter Class Structure*: The filter class contains many methods for handling the different stages needed in the filtering process. Since the filter can be invoked from different locations, the filter class contains several input points that all starts the filtering process. This process contains a series of different tasks that are performed in a particular order, before concluding whether there exists a cross-site scripting injection or not. This includes methods for fetching the input from the request to different methods for comparing this data with either inline scripts, external scripts or on-event handlers, all of which need to be processed differently.

C. Environment

This section will describe the system and tools used when developing the Firefox filter. The operating system used is Arch Linux [42], a lightweight and flexible Linux distribution. For developing and writing the source code, the free and open-source text editor Visual Studio Code (VS Code) [43] was used.

1) *Tools*: Several different tools were utilized during the development of the proposed filter.

a) *Development Software*: When developing computer software, there exist several Integrated Development Environment (IDE) and code editors with a lot of added functionality for helping with software development. For the development

and writing of the source code for this project, a lightweight, free and open source text editor, VS Code [43], was used. VS Code provides the necessary syntax highlighting and autocomplete, while also making it easy to navigate around in the huge Firefox source code. Without adding extra additions to VS Code, it does not handle building and debugging of the Firefox code, which is one of the reasons it is a lightweight editor. For these operations, however, there are more specialized tools that are better suited for the development of Firefox, as Mozilla have their own recommendations and tools available.

b) *Mach*: As mentioned in Theoretical Background II, the tool `mach` [35] is a command-line interface used to start the building, debugging and testing of Mozilla projects, which also was used in the development of this modified version of Firefox. `mach` makes it possible to configure Firefox builds through the usage of a `mozconfig` configuration file [44].

c) *GNU Project Debugger (GDB)*: For debugging, GNU Project Debugger (GDB) [45] was used, a tool that can start programs, make it stop on specified conditions, examine what is happening at runtime and change things in the program as it runs. GDB is a tool that can be invoked using the `gdb` command, but when debugging Firefox it is possible to start GDB through the usage of the `mach` command. After starting the debugging mode, GDB makes it possible to create breakpoints in the code, which lets the debugger inspect the state of the application as it is running.

D. Implementation

This section will describe the implementation of the filter, how it is implemented and integrated into Firefox, also containing details about every part of the filtering process.

1) *Data Flow*: The data flow in Firefox is illustrated in Figure 6, found in Section III-A. This figure is then being expanded in Section IV-B1, Figure 8, where it is shown that the classes `ScriptLoader` and `EventListenerManager` perform several security checks, including using the XSS filter. When the `XSSFilter` class is being invoked from these classes, it first needs to get all the input data from the request. This data is retrieved through the `nsDocument` class. The relevant input data fetched are all the GET- and POST-parameters contained in the request. These parameters are saved in a list, which is then examined further. Every parameter is checked if it contains any potentially malicious content, which in the case of a cross-site scripting attack would be any input that contains some form of script content. This examination is explained further below, in the next section. If the filter identifies any parameters as potentially unsafe, it will compare them to every script entered into the filter class, from the `ScriptLoader` and `EventListenerManager` classes. If any of these scripts are also found in the request, the filter will mark the script as unsafe, which will again notify these classes to not send the detected script to the JavaScript

engine for execution. All the other scripts will be executed as normal.

2) *Examining Input Data:* After fetching all the GET- and POST-parameters from the request, these need to be analyzed for potentially malicious content, which as mentioned above, would consist of any type of script content. It is not a simple task to identify whether or not these parameters contain any actual script content, as there exists many different ways of creating and trying to hide the malicious content of a parameter. A good source of many such attack payloads is OWASP's guidelines "XSS Filter Evasion Cheat Sheet" [16], which contains many examples of injections trying to circumvent typical XSS filtering techniques, including variations of using the `script` tag, `on-event` handlers, as well as other, less used attack vectors. This is why the filter does not actually identify any script content in the parameters before marking them as potentially unsafe, but rather make an assumption based on their contents. If a parameter only contains alphanumeric characters, `[a-z]` `[A-Z]` `[0-9]`, or the underline character, `_`, the parameter is considered safe, and should not be processed further by the filter. These are very common characters that can not be used to execute any scripts, making them safe to include in the response. The reason why the underline character is included is that it is often used in the case of a space in a parameter, which should be considered safe. If there any other characters than the one specified, the filter would include the parameter in further processing, which will be described in more depth in the next section.

3) *Looking for Injections - Matching Algorithm:* If there are any potentially harmful content in the request parameters, for every script received in the response, the filter is running a matching algorithm which tries to identify whether any of these scripts are also contained in any of the parameters. Depending on the type of script received from the response, the filter handles the matching a bit differently. With inline scripts, a comparison of the string representation of the actual script content is done with each and all of the script content from the inline scripts entered through the `ScriptLoader` class. `ScriptLoader` also handles external scripts, in which case it first gets the information about the external URL where the actual script is located before it executes the content inside the script. For the filter, in the case of an external script, it does not do a comparison between the contents of the external script with the parameters, but rather a comparison between the string representation of the external URL and the parameters. As for other attack vectors, like the `on-event` handlers, the same approach as the inline script matching is done. A similarity between the inline and external script matching, however, is that before the actual matching takes place, the content from the scripts and the parameters need to be normalized. This means that these contents might differ slightly, as the parameters content might have changed after going through the HTML parser in Firefox, which again means that some of

the same changes need to be done by the filter for it to detect all injections properly. Several possible factors that need to be addressed when normalizing the contents are listed below, with a basis in the rules from OWASP's filter evasion cheat sheet [16].

a) *Basic evasion techniques:* A basic normalization technique is to not differentiate between upper- and lower-case characters. The script injection `<script src="http://xss.rocks/xss.js"></script>`, which try to load an external script through a different domain, and the slightly different `<script src="http://xss.ROCKS/xss.js"></script>` would thus both be treated as the same injection, as the uppercase characters in the second example would be converted to lowercase. Another basic technique is to use added whitespace or other characters that do not change the behavior of the injected script, but that tries to hide the script from being recognized by filters. An example attack could be the injection `<script>alert (1)</script>`, where additional spaces are included, but where the injection could successfully execute the script content, `alert(1)`. This is related to using different encodings in the injections, which could include more advanced attack payloads.

b) *Different encodings:* It is common for attackers to use different encodings in their attack payloads, by for example using URL encoding [46] for the injected script, which again is a means of hiding the injected string. URL encoding is something that needs to be used in URL's when the URL contains characters outside the American Standard Code for Information Interchange (ASCII) character encoding set, which is why the URL has to be converted into supported ASCII format. This is done by replacing unsafe ASCII characters with a percent sign, `%`, followed by two hexadecimal characters. It is also possible to use this encoding for any input for a website, which means the filter needs to properly decode and identify the encoded data. In this filter's implementation, it is supported by using Mozilla Firefox's own internal class for handling URL's, which also handles decoding of URL encoded data.

c) *Different attack vectors:* The attack vector for injecting XSS attacks used in most examples in this paper, utilize the script tag, `<script>`. It is, however, possible to perform XSS injections by using many other different attack vectors, as explained in Section II-B4. The filter does currently support the `script` tag and every usage of the `on-event` handler, which may be used in combination with many different attack vectors.

4) *Handling of Discovered Script:* If the filter does find a match between a script from the response with a script from the request, it marks that particular script as unsafe and notifies the class that invoked the filter, telling the class that it should not execute this particular script. Even if a script is detected

and blocked, the filter do continue to check all other scripts from the response with the request parameters, as there might be more than one injected script. This is an important aspect of the filter, as it only blocks the actual injected script and not the whole page from loading. By choosing a different solution where the filter is blocking the whole page when an attack is detected, the filter does not need to do any further checking, as you can not execute any more scripts as the page is not being loaded.

5) *Firefox Integration:* This section will briefly describe how the filter class is integrated and how it connects to other parts of Firefox. The filter is implemented as its own class inside Firefox's source code, called `XSSFilter`, making it easy for other components to use the filter when needed. The class is located in the `mozilla/dom/security` folder, which is the same location as where all the Content Security Policy (CSP) related classes reside. The filter is currently being created in places where the filtering functionality is needed, by supplying it with the owning document class, `nsDocument`, in its constructor. As discussed in Section IV-B1, `ScriptLoader` is one of the primary classes that use the filter. Upon creation of the `ScriptLoader` class, it also creates a filter instance with the main document in its constructor. Every time the main document is loading new data, like updated GET- and POST-parameters, the `XSSFilter` instance located in `ScriptLoader` also gets updated, fetching the new request data, before using it in the filtering process every time `ScriptLoader` encounters a script, either inline or external. Another internal class in Firefox, `EventListenerManager`, do also use the `XSSFilter` in a similar manner, but rather than inline and external scripts it takes care of scripts from on-event handlers.

The `XSSFilter` class itself is also accessing other components inside Firefox. To retrieve the GET parameters it has to access the URL from the main document class before using the `URLParams` class for parsing it correctly, making sure the content is properly URL-decoded. As for the POST parameters, the filter gets access to the `nsIHttpChannel` class through the main document, which contains the necessary data for retrieving the parameters, by utilizing different helper classes in Firefox. It also uses several helper classes for a lot of string manipulation, operations like searching for whole strings or single characters, or converting between different types of strings and encodings.

6) *Challenges:* There have been some challenges with the implementation of the filter. Since the filter is being implemented inside an already built software, the Mozilla Firefox web browser, the filter needs to be integrated in a way so that it can cooperate with existing code, data flow and different ways of doing things. Mozilla Firefox is a very huge piece of software, containing many different classes spread across separated modules that talk to each other by using

different means. To properly understand this whole structure and following the data flow proved to be a challenging task, as there were used a lot of different coding principles and internal code for different tasks. String-handling is a good example of how complex the code is, as there exists many different types of strings and as many ways of converting between them and utilizing them correctly.

7) *Unit Testing:* Unit testing is a good way of assuring that separate parts of the code is working as desired. In the case of the filter implementation, the parts containing the examination of input data and the matching algorithms are the most important to test, as these are the parts dealing with the actual filtering process. Several unit tests have been implemented to verify this process, by supplying some sample injected data. As the filter require some special characters to be included in the parameter for it to be checked for in the matching process, several tests have been implemented confirming these character checks. The matching algorithm also have several tests with different injection inputs, verifying that the string matching works correctly. As for testing other parts of the filter, which relies on many different parts of other functions in Firefox, a more complete testing is done in Section V.

V. ANALYSIS AND ASSESSMENT

The filter needs to be evaluated, as explained in Section II, in terms of several different categories. The filter should be tested for how well it protects against XSS attacks and how much it affects the performance of Firefox . An analysis of the filter's implementation, some of the design choices and different limitations are also an important part of the evaluation, as it will highlight what is good and what needs to be improved.

A. Protection Effectiveness

Protection effectiveness is about how well the filter is able to protect against XSS attacks, in particular, Reflected XSS attacks.

1) *Methodology of Testing:* To be able to measure the effectiveness of the filter, it is necessary to conduct testing by doing an examination of a known vulnerable website, as it is not the website's own security features that need to be tested, but the filter's capabilities. One way of making sure this is the case is to implement a sample website, used for the sole purpose of testing the filter. The created website should try to mimic some of the functionality found on other typical websites, as this would provide a better generalization of the filter's overall effectiveness. A common functionality found on a majority of websites is the search field, which is also susceptible to Reflected XSS attacks. The website should, therefore, consist of a search field, which would send the query to a web server, where the response should be a page

containing the input query from the search field. Since the website has no built-in security features, inputting a script into this search field would effectively execute it upon receiving the response. By visiting this vulnerable website through the modified version of Firefox, containing the XSS filter, the filter should be able to both detect and stop the injected script from being executed. This is being tested by conducting an automated test consisting of several different script injections, to see if the filter detects all of the attacks or just a subset of them. The automated test is made possible by the usage of Selenium WebDriver [47], which makes it possible to do direct calls to a specific web browser instance, by using its native support for automation. A simple script will be created that uses Selenium, which takes a list of injections as input, which will then test each of them against the sample vulnerable website. The outcome of this script will be a list of both the successfully injected scripts and the ones that did not get injected.

The script injections that are to be tested, are collected from a variety of sources. An extensive list found on the website gbhackers.com [48], and three different collections gathered from github.com [49] [50] [51]. In total, a list containing 920 unique script injections were created from these sources. This list consists of many different attack vectors targeted at very specific functionality of common websites. Since the sample vulnerable website created is a very simple website, not containing a complex usage of different HTML tags, it is assumed that most of the injections would not be successfully injected. This is why several hundred injections were collected, to make sure that a big enough subset would actually be successfully injected, which could be used in the analysis. For achieving accurate results, the automation testing script would actually need to be executed twice. This process is shown in Figure 9. First, all the injections had to be tested against the vulnerable website *without* the filter enabled. This way, all the injections that are actually working on the vulnerable site, would be recorded in a list created by the testing script. Next, the list of injected scripts would be used to run the testing script another time, this time using a version of Firefox that has the filter enabled. The script would once again create a list containing both the successful injections and the injections stopped by the filter, which then would be used for further analysis. This is done to make sure that the analyzed results are containing actually injectable script content so that it is known that it is the filter that stops the injections, and not something wrong with the injections themselves.

2) Results: When running the automated test as described above, the website without the filter was successfully injected with a total of 138 different script injections. Although many of the injections used similar attack vectors, there were still a good mixture of different attack vectors and encodings used, typically trying to circumvent filtering mechanisms. When using these injections in the version of Firefox containing

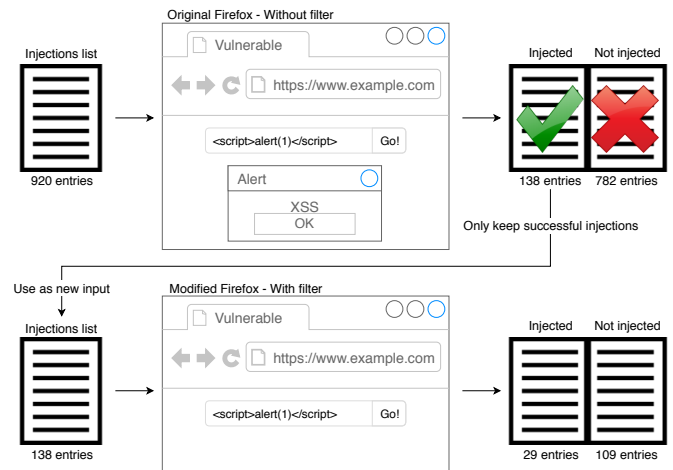


Figure 9: Testing of the implemented filter

the XSS filter, only 29 were successfully injected. The filter did, therefore, block 109 of 138 injections. By examining the results further, it is possible to pinpoint the weaknesses of the filter, which again could be used to improve it.

a) Blocked scripts: Most of the script injections were both detected and blocked by the filter. This included the usage of many different variations of the `script` tag, where the injections were adding other unnecessary characters or using URL encoding trying to circumvent the filter. Because of the filter's location, behind the HTML parser, and the fact that all parameters get URL-decoded, all of these injections were blocked. There were also a lot of usage of `on-event` handlers utilizing similar circumvention techniques. Most of the `on-event` handlers were also blocked, used in combination with different attack vectors like the `img` tag, `svg` tag and `body` tag, since all of these `on-event` handlers had to go through the `EventListenerManager` class, where the filter was invoked from.

b) Injected scripts: As there were a total of 29 successful injections, it is interesting to analyze why the filter did not detect them. Table II contains an overview of the injections, which will be further analyzed here. 16 of 29 of the successful injections used the HTML tag `iframe`, in different forms, utilizing upper/lower capitals, URL encoding and otherwise including different characters to confuse the filter. The `iframe` tag allows web pages to load other web pages inside itself, where the tag also supports the usage of `on-event` handlers. Even though 16 of 138 of the `iframe`-injections got successfully executed, the filter did actually block the instances utilizing `on-event` handlers, as this is well supported by the filter. The filter does not, however, detect `iframe`'s using the `src` attribute, as it is not being invoked in the parts of Firefox that handles the script content inside these tags. 7 other injections were also cases where the attack vectors are not supported, which used the `embed` tag, `svg` tag

and the `object` tag. The last 6 cases, however, used either the `script` tag or `on-event` handlers, but did not get detected. This is because they used various encodings, like HTML entity encoding and base64 encoding, which are not supported by the filter. This is a good example showing that only dealing with the most common URL encoding is not enough, as there exist several other encodings that might be interpreted as script content by the website, that also needs to be considered.

Table II: Testing of the implemented filter

Attack vector	Number	Not supported	Difficulty to fix
iframe	16	attack vector	easy
embed	3	attack vector	easy
object	3	attack vector	easy
svg	1	attack vector	easy
script	2	encoding	moderate
on-event	4	encoding	moderate

3) *Limitations*: There are several limitations regarding the capabilities of the filter, which could be categorized into several categories. Some limitations were related to the actual filtering rules, which means the capability of the filter to detect different types of script injections, using different methods for trying to circumvent the filter. The other types of limitations is related to the different input and output sources supported by the filter, as there are more ways than using script tags and `on-event` handlers to inject script content into websites.

a) *Limitations regarding filtering rules*: As described in Section IV-D3b, the filter did support URL encoded data, which turned out to work really well, stopping several injections. It did not, however, support HTML entity and base64 encoding, which led to script injections being executed in the browser. Support for more different encodings should, therefore, be implemented.

As seen from the results, every injection utilizing the `iframe` tag was successfully injected and executed in the browser, as this was one amongst several injections in which the injected attack vector was not accounted for by the filter. This is a general limitation that the filter simply supports too few attack vectors utilizing different HTML tags. Although the filter supports `on-event` handlers, which is used by a vast amount of HTML tags, these `on-event` handlers are not always necessary to trigger a script for execution, which is why this support needs to be improved.

In Section II-D0b, some limitations of the XSS Auditor filter were discussed, which are tightly related to the limitations of this filter implementation, as they are based on the same

string matching design. Not all of the limitations from Auditor applies though, as this filter does not require the same strict subset of special characters to be present, as Auditor requires. However, the limitations regarding partial string injections are something that has not been addressed in this filter either. If a website have several input fields where its content gets concatenated without proper validation, an attacker might take advantage of this to create a complete injection by splitting the injection into two or more fields. It is worth mentioning that this is a rather special case, as the website needs to have some very specific functionality for this attack to work, but it is still a possibility that should be considered to be addressed.

b) *Limitations regarding request input sources*: Another type of limitation is regarding every input source from the request, which means every source of user modified fields that might enter into a web application. The absolutely most used input sources are the GET- and POST parameters, which are currently the only sources supported by the filter. There are, however, other possible input sources where users could inject malicious content, like for example HTTP headers and cookies. Although these are more special cases, where the web applications need some more specific use-cases, they might still occur, which is why they should be considered to be supported.

B. Performance

The performance of the implemented filter is an important factor for its usefulness. For measuring the performance, Mozilla's own methodology for comparing page load times across browsers [52] was used. This methodology consists of choosing a set of websites that are loaded in Firefox, repeated several times, while measuring the loading time for each page load. This is a process that is automated with the help of Selenium WebDriver [47], which makes it possible to make direct calls to specified browsers using their native support for automation. For this implemented filter, it would be interesting to compare the performance of the modified Firefox instance with the original Firefox instance, which does not include a built-in XSS filter. By using the Selenium WebDriver it is possible to supply both of these instances as options, which means that the testing would be fully automated. As mentioned, it is necessary to have a set of websites to be used for testing. In the case for Firefox's own testing, they chose to pick the 200 most popular websites from the Alexa page rank site [6], because news sites typically contain a lot of trackers.

1) *Methodology of Testing*: For the testing of this filter, news sites are also well suited, as they contain a lot of script content and most often also contains a search field for looking up articles, which is something that is useful for invoking the filter mechanism. For the testing, only a subset of the most popular news sites from the Alexa page rank site were chosen, as not every news site had a working search field. A total of

20 news sites were selected for the testing. It is assumed that most of the top news sites can be considered to be relatively safe, not containing any easy to exploit cross-site scripting vulnerabilities. This does not, however, hinder the filtering mechanism to activate, since the filter would still search the request parameters for potential dangerous contents, and do the comparison between them and the scripts contained in the response. This is done regardless of the existence of any actual vulnerabilities or not, since that is the whole point of the filter, to act as an added layer in the defense in depth strategy trying to stop attacks from potential vulnerabilities.

To make sure the modified browser actually runs the code for the implemented filter, each website was given some input data by using their search fields. The testing was done with two different input data, with the first one simulating a totally legit request that does not contain any script content at all, inputting the query `article`, and the second one containing a simple script, `<script>alert(1)</script>`, simulating a very simple XSS attack. In the first case, by inputting a safe query, the filter would inspect this query and not find any potentially dangerous characters, which means the filter would not need to do any additional processing. In the second case, the same inspection of the query would be done, which would mark the injection as unsafe. After marking it as unsafe, every time a filter would get a script from the response, this script would be matched against the unsafe parameter, trying to identify if the parameter is contained in any of the scripts. The performance difference between the original and the modified browser should be expected to be lower from the first case than the second case, as the filter is doing more work the second query. One thing to notice here is that the filter would most likely not detect an actual attack, as previously assumed that popular news websites are probably protected against simple injection attacks.

2) *Results:* After running the automated test, the result does not suggest any added performance overhead by including the filter. The measured load times were actually so similar that an accurate estimate of how much the filter affected the performance is not possible to measure. Table III illustrates the results, where the unit of the load times are milliseconds. The columns marked "Invalid" means that a web page did not load correctly, which means it got removed when calculating the average load time. In the case of loading web pages with the query `article`, the version containing the filter did actually perform approximately 3.2% faster on average, than Firefox without the filter. In the case of using the query `<script>alert(1)</script>`, the original Firefox version performed approximately 1.7% faster on average. It is worth noting that the results did not contain any huge fluctuations when performing the test, and the biggest difference after calculating the average for each test run was about 362 ms, which was the difference between Run 1a of the original version and Run 2s of the original version.

The difference between the different runs of the modified filter was really small, as seen in the figure. As the total difference between the original and modified versions are also relatively small, the conclusion is that the filter did not add any measurable performance overhead, meaning it achieves very high performance. There are, however, several factors that might have affected the testing, as described in the next section, V-B3. Although, since there were so few fluctuations between the calculated averages, it is assumed that the results reflect the reality fairly well.

Table III: Loading times results, measured in milliseconds

Version:	Original Firefox – Without Filter			Modified Firefox – With Filter		
Query:	article			article		
	Total	Invalid	Avg. per site	Total	Invalid	Avg. per site
Run 1a:	4938542	21	5044,48	4817315	2	4826,97
Run 2a:	4932779	0	4932,78	4817343	2	4827,00
			4988,63			4826,98
Query:	<script>alert(1)</script>			<script>alert(1)</script>		
	Total	Invalid	Avg. per site	Total	Invalid	Avg. per site
Run 1s:	4793049	7	4826,84	4844399	1	4849,25
Run 2s:	4668455	3	4682,50	4830490	0	4826,84
			4754,67			4838,04

3) *Limitations:* There are several factors that might have affected the performance testing, which could mean the results are misleading. When Mozilla did their own performance testing, they used a total of 200 different websites, a number much higher than what was used when testing this filter. Choosing a larger subset of websites for the testing could have given some results reflecting a more average loading time, but the 20 selected websites did achieve a very small variance in the calculated average, so it should not be of much difference if choosing to include any more than this. Some other factors that might have had more impact on the results are fluctuations in the local Internet speed of the testing machine and the fluctuations in the web traffic received by the tested websites at the time of the testing. It is typical that these factors varies throughout the day, depending on the time. The test of the original and modified browser were done consecutively, where each test, where one test contains loading of 1000 websites, took approximately 80 minutes to perform. This is not a very huge time span, meaning these fluctuations should not be considered to be of any huge significance. Another factor that is less likely to have affected the performance is the processing power of the testing machine itself, meaning the CPU of the machine might have been running different tasks when conducting the testing of the different browsers. The testing machine was, however, left alone during the actual testing period, which should result in minimal affection from other tasks running.

These are all limitations that somehow might have affected the testing results, some easier than others to control and minify, which was done to the best of ability. Each of them should not be of any significance, and the results are considered to be

very accurate, but it is still worth mentioning these limitations, as is often small variances in the results which should be tried to be explained.

C. Implementation

It is also interesting to analyze how well the filter itself is implemented, in terms of how well it is integrated into Firefox, and how it affects the usage of Firefox other than the already measured performance.

1) Conform to Mozilla Firefox's Internal Coding Standards: Mozilla Firefox has strict guidelines for how things should be integrated into the browser, a coding standard for everything from simple formatting to the usage of different parts from the code. The implemented filter has tried to comply to these rules, by following the general coding standards, particularly regarding the handling of strings [53], as string matching has been a major part of the filter mechanism. Getting access to other parts of the code, parsing data correctly, exception handling, and testing are other examples of good implementation regarding Mozilla's coding principles. There is, however, one aspect of the implementation that is not being integrated well enough for being part of a release version of Firefox. This is the fact that the filter is not utilizing the concept of script security and the usage of principals, as explained in Section III-B.

2) Blocking Technique: When detecting a potential XSS attack, the filter should be able to act upon it and block the script injection. There are several ways of doing this blocking, as mentioned in Section IV-A2, it is possible to only block the injected script or the whole web page. Both of these techniques have their advantages and disadvantages, which are being discussed here.

a) Partial blocking: One of the reasons for blocking only the injected script is that it would interfere less with a user's normal browsing of web pages, as the user could still use the other parts of the web page, which are not affected by the injected script. This is also a huge advantage in the case of a false-positive, again as the user gets less interrupted, as only a subpart of the page gets blocked.

b) Blocking the whole page: There are, however, some disadvantages when choosing to only block parts of the page. When the filter detects an attack, it is not unexpected that an attacker might have combined several techniques and parts when injecting the script into the website, hoping that one of the included parts of the script would be able to circumvent the filter. Hopefully, the filter would be advanced enough to properly detect and block all the parts of the injection, but it might be some special conditions that the filter does not account for, leading to a successful attack. This is one of the reasons why it might be a better approach, when only concerned with security, to block the whole web page from loading when an attack is detected, as the detected attack might just be part of a bigger

attack. Another possibility for an attacker is to trick the filter to not block an injected script, but to block some important security feature that is actually needed by the attacked website itself. An example is a website that requires the JavaScript file `security.js` for its security features to work, which will be included in the response when requesting the website. Since the filter compares script content from the request with script content from the response, an attacker might inject a script containing the same filename, `security.js`, which would then be detected by the filter as an attack, as the file is both in the request and the response of the website. This would then disable the websites security features, which means the attacker could create an injection that combines the file `security.js` with some other malicious script executing an attack. Since the filter actually detected an attack, it would be better to block the entire web page from loading, as this would prevent this issue altogether.

3) Usability: The implemented filter does not currently support any interaction from or with the user of Firefox, which is something that should be considered, as more control of and information about the filter's behavior could be beneficial to websites and Firefox's users.

a) Choosing blocking technique: As there are clearly advantages and disadvantages with both the blocking techniques, it is possible to make this a decision for websites to take, by utilizing the `X-XSS-Protection` HTTP response header [54]. This is a header currently supported by most of the major web browsers, Chrome, Safari, Internet Explorer and Edge, and makes it possible to decide how the browser should act when they detect XSS attacks. There are four possible values for the `X-XSS-Protection` header. Setting it to 0 will disable the filter and 1 will enable it and only remove the dangerous parts. By using `1; mode=block`, the filter will be enabled and the whole web page will be blocked. A last option is using `1; report=<reporting-uri>`, which will only remove the dangerous parts and use a feature from CSP where the violation is reported and sent to the specified URL.

b) Violation feedback: Another functionality missing from the implementation, something the implemented CSP feature already has, is the ability to properly notify users of a violation. In the case of an XSS violation, this would be when the filter detects and/or blocks the attack, depending on what the previously mentioned `X-XSS-Protection` header is set to. This header, did as mentioned above, support a reporting feature, where details of the violation would get sent to a specified URL. However, a violation notice should also be indicated to the users of Firefox, regardless of the reporting feature of the `X-XSS-Protection` header. In the filter's current state, these violation details are only shown in a special console meant for the developers of Firefox itself, and not in the developer console accessible to normal users of Firefox. The details shown to the users does not have to contain every detail

about the violation, but an indication of what has happened should be displayed.

VI. CONCLUSION

Cross-Site Scripting (XSS) vulnerabilities continue to be one of the most critical web security threats among today's web applications, despite the large quantity of research, proposals and solutions being published and implemented [5]. This is a type of vulnerability that mainly and directly compromise the end-users of web applications, which means they need additional protection. All of the major web browsers have taken action by implementing several protection mechanisms defending against these XSS attacks. Since XSS is such a complex vulnerability, there is not a single protection mechanism that will stop all of the attacks, but rather a strategy of having several mechanisms that together provide the best protection, utilizing the defense-in-depth strategy. All of the major web browsers have included a built-in filter for XSS protection as one of these counter measures, except the second most used, Mozilla Firefox, which have neglected to include such a feature. As seen from the lacking effectiveness of the most comprehensive protection mechanism in Firefox, CSP, as discussed in Section II-C, the need for a built-in XSS filter in Firefox is evident, considering the prevalence and consequence of these attacks.

This paper has made a proposal and implementation for such a filter, which is built-in and integrated into Firefox. The filtering principles for the filter was based on the filter used in Google's Chrome browser, XSS Auditor, which utilizes an advantageous placement inside the web browser, achieving both good protection and high performance. After doing several tests of the implemented filter, findings suggest that the filter did perform very well in protecting against a wide variety of script injections, which contained different attack vectors utilizing several methods trying to circumvent the filtering mechanism. Adding and removing characters, using URL encodings and different on-event handlers were efficiently blocked by the filter. There were, however, some limitations regarding different types of encodings and a lack of support for some attack vectors, which are something that needs to be added before the filter could be considered sufficient for every-day usage. Performance-wise, the filter did not show any measurable difference compared to the version of Firefox without the filter. By not having any huge performance overhead means that adding small additions for fixing the limitations mentioned should not incur significantly more overhead, as the most demanding filtering mechanisms are already implemented.

The modified version of Firefox containing the filter do, therefore, already provide much better protection than the original version of Firefox. Even though there are limitations that needs to be addressed for it to be a considered a fully

fledged solution, it already serves as an important layer in the defense-in-depth strategy, providing a little extra to the much desired protection that is needed for XSS vulnerabilities.

VII. FURTHER WORK

As discussed in Section V, the implemented filter still has room for improvements considering its protection effectiveness. The areas for improvements are regarding input sources, attack vectors, support for more encodings and integration with existing Firefox code. Most of these improvements should be rather trivial to implement. Firefox's internal code has easy access to other input sources data, like the most relevant, which are HTTP headers. In the case of attack vector support, the already supported attack vectors only needed about two lines of code for them to be covered, so it should be as trivial to add support to other vectors, like the `iframe`, `embed`, `svg` and `object` tags, as mentioned in Section V-A2b. The only challenge with these is to identify the location inside the Firefox code where they are being processed, as they might be handled in vastly different areas in the code. Support for more encodings should also not be too difficult to achieve, as there exist good documentation covering how different encodings work, and the fact that the filter class is structured in such a way that it is easy to add more advanced filtering rules. The most challenging task would be to better integrate the filter into the existing Firefox code, to comply with all the security principals and coding standards that are required by Mozilla. Another improvement could be to implement support for the X-XSS-Protection header, which would let websites themselves decide if they want to use it or not.

REFERENCES

- [1] A. Vikne and P. Ellingsen, "Client-Side XSS Filtering in Firefox," in SOFTENG 2018, The Fourth International Conference on Advances and Trends in Software Engineering, April 2018, pp. 24–29.
- [2] OWASP Foundation, "Owasp top 10 - 2017 the ten most critical web application security risks," accessed: 2017-12-27. [Online]. Available: https://www.owasp.org/images/7/72/OWASP_Top_10-2017_%28en%29.pdf.pdf
- [3] WhiteHat Security, Inc., "2017 whitehat security application security statistics report," 2017, accessed: 2017-12-21. [Online]. Available: <https://info.whitehatsec.com/rs/675-YBI-674/images/WHs%202017%20Application%20Security%20Report%20FINAL.pdf>
- [4] Bugcrowd Inc., "2017 state of bug bounty report," 2017, accessed: 2018-01-09. [Online]. Available: <https://pages.bugcrowd.com/hubfs/Bugcrowd-2017-State-of-Bug-Bounty-Report.pdf>
- [5] I. Hydar, A. B. M. Sultan, H. Zulzalil, and N. Admodisastro, "Current state of research on cross-site scripting (XSS)—A systematic literature review," *Information and Software Technology*, vol. 58, 2015, pp. 170–186.
- [6] Alexa Internet, Inc., "The top 500 sites on the web," January 2018, accessed: 2018-01-15. [Online]. Available: <https://www.alexa.com/topsites>
- [7] StatCounter, "Desktop browser market share worldwide," May 2018, accessed: 2018-05-09. [Online]. Available: <http://gs.statcounter.com/browser-market-share/desktop/worldwide>

- [8] Mozilla Developer Network, "Confidentiality, Integrity, and Availability," April 2018, accessed: 2018-04-19. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Security/Information_Security_Basics/Confidentiality,_Integrity,_and_Availability
- [9] M. Alvarez, N. Bradley, P. Cobb, S. Craig, R. Iffert, L. Kessem, J. Kravitz, D. McMillen, and S. Moore, "IBM X-Force Threat Intelligence Index 2017 The Year of the Mega Breach," IBM Security,(March), 2017, pp. 1–30.
- [10] Mozilla Developer Network, "HTTP headers," April 2018, accessed 2018-05-13. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers>
- [11] —, "Referer," June 2017, accessed 2018-05-13. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Referer>
- [12] —, "User-Agent," June 2017, accessed 2018-05-13. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/User-Agent>
- [13] S. Di Paola and G. Fedon, "Subverting ajax," 2006.
- [14] Facebook Inc., "Httponly," August 2017, accessed: 2018-03-23. [Online]. Available: <https://www.facebook.com/help/246962205475854>
- [15] Mozilla Developer Network, "DOM on-event handlers," Jan 2018, accessed 2018-05-24. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Guide/Events/Event_handlers
- [16] OWASP Foundation, "Xss filter evasion cheat sheet," October 2017, accessed: 2017-12-27. [Online]. Available: https://www.owasp.org/index.php/XSS_Filter_Evasion_Cheat_Sheet
- [17] —, "Xss (cross site scripting) prevention cheat sheet," October 2017, accessed: 2018-01-24. [Online]. Available: [https://www.owasp.org/index.php/XSS_\(Cross_Site_Scripting\)_Prevention_Cheat_Sheet](https://www.owasp.org/index.php/XSS_(Cross_Site_Scripting)_Prevention_Cheat_Sheet)
- [18] Mozilla Developer Network, "Content security policy (csp)," January 2018, accessed: 2018-01-24. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/CSP>
- [19] The World Wide Web Consortium, W3C, "Content security policy level 2," December 2016, accessed: 2018-01-11. [Online]. Available: <https://www.w3.org/TR/2016/REC-CSP2-20161215/>
- [20] L. Weichselbaum, M. Spagnuolo, S. Lekies, and A. Janc, "Csp is dead, long live csp! on the insecurity of whitelists and the future of content security policy," in Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016, pp. 1376–1387.
- [21] Mozilla Developer Network, "Same-origin policy," March 2018, accessed: 2018-03-21. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/Security/Same-origin_policy
- [22] OWASP Foundation, "Httponly," August 2017, accessed: 2018-03-21. [Online]. Available: <https://www.owasp.org/index.php/HttpOnly>
- [23] S. Gupta and B. B. Gupta, "Cross-Site Scripting (XSS) attacks and defense mechanisms: classification and state-of-the-art," International Journal of System Assurance Engineering and Management, vol. 8, no. 1, 2017, pp. 512–530.
- [24] D. Bates, A. Barth, and C. Jackson, "Regular expressions considered harmful in client-side xss filters," in Proceedings of the 19th international conference on World wide web. ACM, 2010, pp. 91–100.
- [25] G. Maone, "NoScript - JavaScript/Java/Flash blocker for a safer Firefox experience! - features - InformAction," accessed: 2017-12-28. [Online]. Available: <https://noscript.net/features>
- [26] B. Stock, S. Lekies, T. Mueller, P. Spiegel, and M. Johns, "Precise client-side protection against dom-based cross-site scripting," in USENIX Security Symposium, 2014, pp. 655–670.
- [27] Mozilla Developer Network, "eval," April 2018, accessed 2018-05-29. [Online]. Available: https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Global_Objects/eval
- [28] D. Ross, "Ie8 security part iv: The xss filter," July 2008, accessed: 2018-01-11. [Online]. Available: <https://blogs.msdn.microsoft.com/ie/2008/07/02/ie8-security-part-iv-the-xss-filter/>
- [29] Mozilla Developer Network, "History of the Mozilla Project," April 2018, accessed: 2018-04-04. [Online]. Available: <https://www.mozilla.org/en-US/about/history/details>
- [30] —, "An introduction to hacking mozilla," March 2017, accessed: 2017-12-28. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/An_introduction_to_hacking_Mozilla
- [31] —, "Introduction," September 2014, accessed: 2017-12-28. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XUL/Tutorial/Introduction>
- [32] —, "Gecko faq," September 2015, accessed: 2017-12-28. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Gecko/FAQ>
- [33] —, "How mozilla's build system works," December 2017, accessed: 2018-02-14. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Build_Instructions/How_Mozilla_s_build_system_works
- [34] GNU/Free Software Foundation, "Gnu make," May 2016, accessed: 2018-02-14. [Online]. Available: <https://www.gnu.org/software/make/>
- [35] Mozilla Developer Network, "mach," December 2017, accessed: 2018-02-14. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/mach
- [36] I. Black Duck Software, "The Mozilla Firefox Open Source Project on Open Hub: Languages Page," April 2018, accessed 2018-05-22. [Online]. Available: https://www.openhub.net/p/firefox/analyses/latest/languages_summary
- [37] Mozilla Developer Network, "Mozilla Source Code Directory Structure," Jan 2018, accessed 2018-05-22. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Source_Code/Directory_structure
- [38] —, "HTML parser threading," March 2013, accessed 2018-05-30. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Gecko/HTML_parser_threading
- [39] —, "Script Security," Aug 2016, accessed 2018-05-24. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Gecko/Script_security
- [40] —, "Browser Extensions," March 2018, accessed 2018-05-27. [Online]. Available: <https://developer.mozilla.org/en-US/Add-ons/WebExtensions>
- [41] —, "Browser support for JavaScript APIs," May 2018, accessed 2018-05-27. [Online]. Available: https://developer.mozilla.org/en-US/Add-ons/WebExtensions/Browser_support_for_JavaScript_APIs
- [42] "Arch Linux," Jan 2018, accessed 2018-05-30. [Online]. Available: https://wiki.archlinux.org/index.php/Arch_Linux
- [43] Microsoft, "Visual Studio Code - Code Editing. Redefined," accessed 2018-05-30. [Online]. Available: <https://code.visualstudio.com/>
- [44] Mozilla Developer Network, "Configuring Build Options," March 2018, accessed 2018-05-30. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Developer_guide/Build_Instructions/Configuring_Build_Options
- [45] GNU/Free Software Foundation, "Gdb: The gnu project debugger," February 2018, accessed: 2018-03-02. [Online]. Available: <https://www.gnu.org/software/gdb/>
- [46] R. D. W3Schools, "HTML URL Encoding Reference," May 2018, accessed: 2018-05-02. [Online]. Available: https://www.w3schools.com/tags/ref_urlencode.asp
- [47] S. Project, "Selenium WebDriver - Selenium Documentation," April 2018, accessed 2018-05-02. [Online]. Available: https://www.seleniumhq.org/docs/03_webdriver.jsp

- [48] Balaji N., "Top 500 most important xss script cheat sheet for web application penetration testing," May 2018, accessed 2018-05-30. [Online]. Available: <https://gbhackers.com/top-500-important-xss-cheat-sheet/>
- [49] Thapa, Prabesh, "Xss-payloads," Aug 2016, accessed 2018-05-30. [Online]. Available: <https://github.com/Pgaijin66/XSS-Payloads/blob/master/payload.txt>
- [50] FuzzDB Project, "xss-other.txt," Oct 2016, accessed 2018-05-30. [Online]. Available: <https://github.com/fuzzdb-project/fuzzdb/blob/master/attack/xss/xss-other.txt>
- [51] —, "xss-rsnake.txt," May 2016, accessed 2018-05-30. [Online]. Available: <https://github.com/fuzzdb-project/fuzzdb/blob/master/attack/xss/xss-rsnake.txt>
- [52] D. Strohmeier, P. Dolanjski, "Comparing browser page load time: An introduction to methodology," November 2017, accessed: 2018-01-15. [Online]. Available: <https://hacks.mozilla.org/2017/11/comparing-browser-page-load-time-an-introduction-to-methodology/>
- [53] Mozilla Developer Network, "Mozilla internal string guide," April 2018, accessed 2018-05-25. [Online]. Available: https://developer.mozilla.org/en-US/docs/Mozilla/Tech/XPCOM/Guide/Internal_strings
- [54] —, "X-XSS-Protection," Feb 2018, accessed 2018-05-28. [Online]. Available: <https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/X-XSS-Protection>

Dynamic Programming Approach to Retrieving Similar Candlestick Charts for Short-Term Stock Price Prediction

Yoshihisa Udagawa

Computer Science Department, Faculty of Engineering,
Tokyo Polytechnic University
1583 Iiyama, Atsugi, Kanagawa, Japan
e-mail: udagawa@cs.t-kougei.ac.jp

Abstract— The paper describes a method for a short-term stock price prediction based on candlestick chart techniques that are popular among stock traders using technical analysis. While the techniques have long history, there is still no consistent conclusion on the predictability of the techniques. We focus on the fact that a trend of stock prices often continues after intervals of several days because stock prices tend to fluctuate according to announcements of important economic indicators, economic and political news, etc. Typically, stock price movements in the period without important news are small, resulting in generating a series of noisy candlesticks. To cope with the noisy candlesticks, this paper focuses on a dynamic programming algorithm that allows us to perform partial matches on sequences of stock prices. We propose a model consisting of six parameters for retrieving similar candlestick charts in order to take into account where the stock price occurs in high/low price zones, in addition to a price change and a length of candlestick body. Experiments are performed on the daily NASDAQ composite index. We choose the daily time frame since important news that affects stock prices occurs on a daily basis. The statistics of the candlesticks are calculated to determine the parameter values of the proposed model based on the average and the standard deviation. Experimental results show that the proposed method is effective in predicting both uptrend and downtrend. Strictly, the prediction of the downtrend is a little bit difficult than that of the uptrend, probably reflecting the fact that the NASDAQ stock market is constantly growing.

Keywords—Stock price prediction; Technical analysis; Candlestick charts; Longest common subsequence; Statistics of candlesticks.

I. INTRODUCTION

This research paper is based on the previously reported contribution on a candlestick chart retrieval algorithm for predicting stock price trend [1]. We provide details on implementation of the dynamic programming algorithm to eliminate noisy candlesticks that often occur when there is no noteworthy economic news. While experiments are performed on the daily Nikkei stock average (Nikkei 225) in the previous paper, this paper presents experimental results on the NASDAQ composite index [2] for showing the applicability of the algorithm.

Predicting the direction of future stock prices is a challenging topic in many fields including trading, finance,

statistics and data mining in computer science. The motivation is to predict the direction of future prices so that stocks can be bought and sold at profitable positions. Fundamental analysis and technical analysis are two primary approaches to making investment decisions for successful stock trading [3].

Fundamental analysis is an approach involving a study of company fundamentals such as revenues and expenses, business concept and competition, and so on. To forecast future stock prices, fundamental analysis combines economic, industry, and company analysis to evaluate a stock's current fair value and forecast future value. Because of this analyzing processes, most people believe that fundamental analysis is mainly suitable for long-term prediction.

Meanwhile, technical analysis is a method of predicting the future direction of a stock price by studying historical stock price patterns. A technical analysis presumes that those price patterns tend to repeat themselves in the future. One of the important types of technical analysis is candlestick chart patterns [4]. The candlestick chart patterns provide short-term predictions for traders to make buy or sell decisions. While most techniques use patterns of stock prices during more than ten days, the candlestick charting technique focuses on patterns among several days of candlesticks formulated by opening, high, low, and closing prices within a specific time frame, such as minute, hour, day or week. Dozens of candlestick chart patterns are identified to be signals of up, down, and sideways of trend directions. These patterns consist of a single candlestick or a combination of multiple candlesticks usually less than four. In fact, the technique acts as a leading indicator with its capability to provide trading signals earlier than other technical indicators. It is also used by some real time technical service providers to provide quick signals for market's sentiments [5].

The candlestick charting technique probably began sometime after 1850 [4]. Despite of its long history and popularity, mixed results are obtained in the studies on candlestick charting. Negative conclusions to the predictability of candlesticks are reported [6]-[8], while positive evidences are provided for several candlestick chart patterns in experiments using the U.S., European and Asian stock markets [9]-[15].

It is also pointed out that candlestick chart pattern recognition is subjective [4]. The candlestick chart patterns

are often qualitatively described using words and illustrations. The studies [6]-[15] adopt definitions using a series of inequalities with different parameters that specify candlestick patterns. Numerical definitions of candlestick patterns are still controversial issues.

In addition, the previous study [1] mentions that the candlestick patterns do not occur in time series in a strict sense because stock price fluctuation continues after intervals of several days depending on announcements of important economic indicators, economic and political news, etc. Because of these characteristics, the candlestick chart patterns are deemed to bring controversial results on predictability regarding future market trends even short-term prediction.

The aim of the study is to estimate the predictability of candlestick patterns for future stock price trends. The proposed algorithm is applied to the daily NASDAQ composite index, while the daily Nikkei 225 (Nikkei stock average) is used in the previous study [1]. Daily historical stock prices are used because important news that affects stock prices occurs on a daily basis, and we plan to relate chart patterns to economic and political news in the future study.

The contributions of this paper are as follows:

- (I) We propose a novel model for retrieving candlestick patterns that includes six attributes. The model takes account of 5-day moving average and 25-day moving average to decide whether the patterns occur in high or low price zones of a stock, which is original to the best of our knowledge,
- (II) The values of the attributes of the proposed model are estimated based on statistical analysis of candlesticks so that the experimental results are evaluated in terms of statistics hopefully being applicable to world markets,
- (III) The LCS (Longest Common Substring) algorithm [16] is improved to make an optimal matching of candlestick patterns that contain noisy candlesticks,
- (IV) The proposed model devises a graphical representation to make evaluation of the retrieval results easy to depict the predictability for short-term trends.

The remainder of the paper is organized as follows. Section II gives some of the most related work. Section III describes backgrounds of the candlestick chart. Section IV proposes a model for retrieving similar candlestick charts. An augmented dynamic programming technique is adopted to implement the proposed model. Section V presents experimental results on both uptrend and downtrend of stock prices. Section VI concludes the paper with our plans for future work.

II. RELATED WORK

The principles of technical analysis were derived from the observation of financial markets over hundreds of years. In Europe, the dawn of technical analysis appeared in Joseph de la Vega's accounts of the Dutch markets in the 17th century. In Asia, the candlestick charting technique emerged during early 18th century, and probably established sometime after 1850 [4]. In the U.S., the Dow Theory traced

back to 1884. However, the technical analysis is widely dismissed by academics in the 1970s. In particular, it is rejected by the weak form of the EMH (Efficient Market Hypothesis) formulated by Fama [17]. The EMH states that stock prices adjust rapidly to the arrival of new information, there is no way to outperform the market average. The studies of the last two decades show controversial results for supporting the EMH and opposing it.

Some studies [6]-[8] find that the candlestick charting is useless based on the experiments using the stock exchange markets' data in the U.S., Japan and Thailand. Horton [6] examines candlestick patterns for 349 stocks finding little value in the use of them. Marshall, Young, and Cahan [7] investigate the profitability of candlestick trading strategies using nine scenarios with different buy-and-hold strategies. They conclude that trading signals generated by the candlestick technical analysis do not have profitable forecasting power on the DJIA (Dow Jones Industrial Average) and the Japanese market, which is consistent with the EMH. Tharavanij, Siraprasit, and Rajchamaha [8] investigate the profitability of uptrends and downtrends of candlestick patterns consisted of one-day, two-day, and three-day candle sticks. Based on experiments using stock data in the SET (Stock Exchange of Thailand), they conclude that any candlestick patterns cannot reliably predict market directions even with filtering method using well-known stochastic oscillators [4].

Other studies conclude that applying certain candlestick patterns is profitable at least for short-term trading [9]-[13]. Caginalp and Laurent [9] favorably evaluate the predictive power of eight three-day reversal candlestick patterns using S&P500 stock price data from 1992 to 1996. They propose to define candlestick patterns and price trend as a set of inequalities using opening, high, low, and closing prices. These inequalities are taken over in the later studies. Goo, Chen, and Chang [10] define 26 candlestick patterns using modified version of inequalities that are proposed by Caginalp and Laurent. They examine these patterns on stock data of Taiwan markets, and conclude that some of the candlestick trading strategies are valuable for investors.

Chootong and Sornil [11] propose a trading strategy combining price movement patterns, candlestick chart patterns, and trading indicators. A neural network is employed to determine buy and sell signals. Experimental results using stock data of the SET market show that the proposed strategy generally outperforms the use of traditional trading methods based on indicators.

Lu, Chen, and Hsu [12] apply candlestick trading strategies to the U.S. market data with several trend definitions. They claim that the trading strategies appear to possess predictive power on a price trend. Specifically, they indicate that three-day reversal patterns are profitable when the transaction cost is set at 0.5%. They also find that holding strategies play an important role to improve the effectiveness of candlestick charting.

Zhu, Atri, and Yegen [13] examine the effectiveness of five different candlestick reversal patterns in predicting short-term stock movements using stock data of two Chinese markets. The results of statistical analysis suggest

that the candlestick patterns work out in predicting price trend reversals.

The following two studies conclude cautious results. Martinsson and Liljeqvist [14] give a set of inequalities to define candlestick patterns including the length of body, the change of price and the length of shadows. They evaluate the impact of the candlesticks trading strategies and the RSI (Relative Strength Index) to evaluate the trend of the market. While they have positive results on the Swedish OMXS30 exchange, they have negative results on the London FTSE100 exchange. Chin, Jais, Balia, and Tinggi [15] examine the predictive power of candlestick continuation patterns, which predict to continue in the direction of original trend, in a Malaysian stock market from 2000 to 2014. They conclude that only one downtrend continuation pattern shows significant predictive power during the 5-day holding period.

The studies [6]-[15] translate the candlestick verbal and visual descriptions into numeric formulas in order to be used in an algorithm. However, they fail to consider zones where the candlestick patterns of interest occur. The interpretation of candlestick patterns depends on the price zone, e.g., high, low, neutral. For example, the morning star pattern generally suggests an uptrend when it occurs in a low price zone. However, the morning star pattern is deemed to be less predictable when it occurs in a high price zone than it occurs in a low price zone.

Most importantly, the studies [6]-[15] do not discuss neutral or noisy candlesticks that often take place in charts because stock prices are apt to depend on important economic and political news and events.

III. CANDLESTICK CHART AND PATTERNS

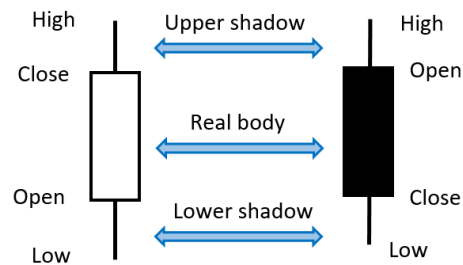
This section introduces the formation of a candlestick. Candlestick patterns are a combination of one or more candlesticks [4]. Samples of well-known candlestick chart patterns are depicted. Because the candlestick patterns are described in natural language and illustrations, there are criticisms on their use for trend prediction by a computer.

A. Formation of Candlestick

A daily candlestick line is formed with the market's opening, high, low, and closing prices of a specific trading day. Figure 1 represents the image of a typical candlestick. The candlestick has a wide part, which is called the "real body" representing the range between the opening and closing prices of that day's trading.

If the closing price is above the opening price, then a white candlestick with black border is drawn to represent an uptrend candlestick. If the opening price is above the closing price, then a filled candlestick is drawn. Normally, black color is used for filling the candle to represent a downtrend candlestick.

The thin lines above and below the body represent the high/low ranges. These lines are called "shadows" and also referred to as "wicks" and "tails." The high is marked by the top of the upper shadow and the low by the bottom of the lower shadow.



(A) Candlestick for price up (B) Candlestick for price down

Figure 1. Candlestick formation.

Typically, a stock's opening price is not identical to its prior day closing price. This is because the time during which stock market is closed changes investor's emotions and expectations for the stock markets with different interpretations of economic news of the day and stock fluctuations.

B. Samples of Candlestick Patterns

Dozens of candlestick patterns are identified and become popular among stock traders [4]. These patterns have colorful names like *morning star*, *evening star*, *three white soldiers*, and *three black crows*.

Figure 2 shows the *morning star* pattern which is considered as a major reversal signal when it appears in a low price zone or at the bottom. It consists of three candles, i.e., one short-bodied candle (black or white) between a preceding long black candle and a succeeding long white one. The pattern shows that the selling pressure that was there the day before is now subsiding. The third white candle overlaps with the body of the black candle showing a start of an uptrend reversal. The larger the white and black candles, and the higher the white candle moves, the larger the potential reversal. The opposite version of the *morning star* pattern is known as the *evening star* pattern which is a reversal signal when it appears in a high price zone or at the end of an uptrend.

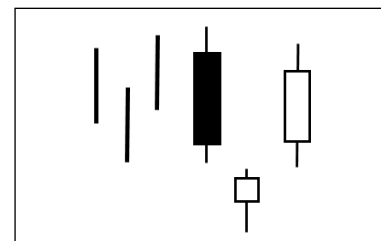


Figure 2. Morning star pattern.

Figure 3 shows the *three white soldiers* pattern which is interpreted as a strong indication of an uptrend market reversal when it appears in a low price zone. It consists of three long white candles that close progressively higher on each subsequent trading day. Each candle opens higher than

the previous opening price and closes near the high price of the day, showing a steady advance of buying pressure.

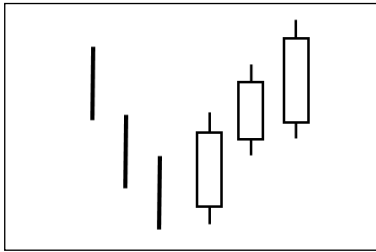


Figure 3. Three white soldiers pattern.

The opposite of the three white soldiers pattern is known as the *three black crows* pattern which is interpreted as a downtrend signal of market trend.

C. Criticism of Candlestick Patterns

The major criticism of the candlestick chart patterns is that the patterns are qualitatively described with words, such as “long/short candlesticks,” “higher/lower trading,” “strong/weak signal,” supported by some illustrations [4]. What percentage of price change does “long/short” mean? Without modeling the candlestick patterns in a way that a computer can process and performing experiments comprehensively based on the statistics of candlesticks, arguments on the effectiveness of chart patterns would not come to an end.

Secondly, the candlestick chart patterns do not deal with market liquidity. Liquid market refers to a market in which there are many buyers and sellers and in which transactions of stocks rapidly take place. In a liquid market, stock price trend to change relatively small. The analogy holds for an illiquid market. It seems necessary to take account of the market's liquidity to improve the predictability of the candlestick chart patterns. We use statistics of candlesticks of market under study to cope with the first and second criticisms.

Finally, the candlestick chart patterns are described under the assumption that candlestick will occur consecutively, which is often not true for actual stock price movements. This study proposes an algorithm using dynamic programming technique for retrieving similar candlestick charts. The algorithm is designed to provide optimal matching between two given price sequences including several noisy candlesticks. The function of eliminating noisy candlesticks distinguishes this study from other ones.

IV. PROPOSED MODEL FOR RETRIEVING CANDLESTICK PATTERNS

This section describes a model for retrieving similar candlestick charts. Following a problem definition, the principle of eliminating noisy candlesticks are described. A dynamic programming technique is used to implement the proposed model. Statistics of candlesticks are calculated in order to estimate the six parameters of the proposed model.

A. Parameters Featuring Candlestick Patterns

As a preliminary stage of study, experiments only using the closing prices and the length of real bodies are performed. The experiments simply correspond to the conditions of the candlestick chart patterns [4]. The results are discouraging. Although mined stock price sequences are similar before the specified period of the reference date, trends of the sequences after the reference date are seemed to be random. Analyses of the results show that the randomness occurs due to the relative position among the stock price, the 5-day moving averages, and the 25-day moving averages.

Based on the results of the preliminary experiments, we propose a model for retrieving similar candlestick charts. Figure 4 depicts the model that consists of the six parameters as follows:

- (1) Change of prices w.r.t previous closing price,
- (2) Length of candlestick body,
- (3) Difference from 5-day moving average,
- (4) Difference from 25-day moving average,
- (5) Slope of 5-day moving average,
- (6) Slope of 25-day moving average.

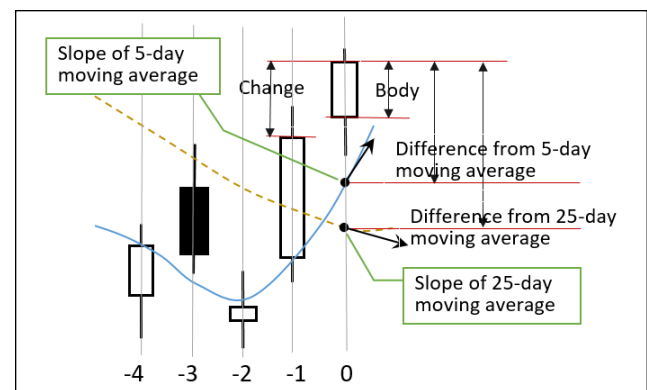


Figure 4. Candlestick pattern retrieval model.

The proposed model is unique because it uses two moving averages and their slopes, while the previous studies [6]-[15] do not deal with them. Relative position among a stock price, 5-day moving average, and 25-day moving average is significant to identify the zone where the candlestick pattern under consideration occurs, which is vital information for applying the candlestick pattern. The slopes of the moving averages are also important to identify their trends, e.g., an uptrend, a downtrend or a sideways (flat).

B. Problem definition and approach to solution

Let r_i ($1 \leq i \leq m$) and t_j ($1 \leq j \leq n$) represent candlesticks r_i and t_j which are defined by a vector of six parameters mentioned in (1) - (6). Let $R = (r_1, \dots, r_i, \dots, r_m)$ denote a reference candlestick pattern, and $T = (t_1, \dots, t_j, \dots, t_n)$ denote a test candlestick pattern with lengths of m and n , respectively. The problem we are dealing with is to find maximum matching of candlesticks r_i and t_j while eliminating unmatched ones.

Dynamic programming is a computer programming method that finds optimal solutions of a complicated problem. There are dozens of dynamic programming algorithms that implement optimal matching between sequences under certain criteria. Among them, we focus on finding the longest price sequence in common between stock price sequences. Because the LCS algorithm essentially finds the longest matched elements while discarding unmatched elements of sequences [16], the LCS algorithm fulfills our intention of eliminating noisy candlesticks.

The algorithm apparently satisfies the requirements of finding the longest price sequence. However, it is originally developed for strings of characters. The fact motivates us to modify the LCS algorithm to handle numeric sequences.

C. nLCS: LCS for Numerical Subsequences

The LCS algorithm is originally developed for character strings. Finding the LCS between two strings is described as follows. Given two strings, find the longest character subsequence that presents in both of them. Characters of the subsequence appear in the same relative order, but not necessarily contiguous. Figure 5 depicts the LCS of the two strings “246612” and “3651.” Since elements of sequences are interpreted as characters that require an exact match, the LCS is “61.”

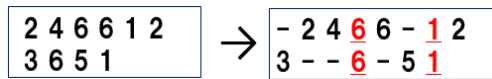


Figure 5. The LCS of two character sequences, “246612” and “3651.”

It is rather easy to improve the LCS algorithm to deal with numerical sequences (nLCS) by interpreting each element as a number and using a tolerance given by a user. If the difference of two numbers is not greater than the given tolerance, then the two numbers are regarded as the same. For example, let the tolerance be set to one, and the two number sequences be “246612” and “3651.” The nLCS are “2661” and “3651” as shown in Figure 6.

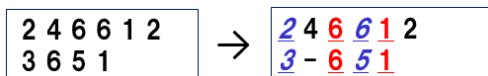


Figure 6. The nLCS of the two number sequences “246612” and “3651” for the tolerance of one.

The LCS and nLCS are formally defined as follows.

LCS algorithm: Let the input sequences be $X = (x[1], \dots, x[m])$ of length m and $Y = (y[1], \dots, y[n])$ of length n . Let $D[i, j]$ denote the length of the longest common subsequence of $x[i]$ and $y[j]$ for $0 \leq i \leq m$ and $0 \leq j \leq n$.

- If either sequence or both sequences are empty, then the LCS is empty, i.e., $D[i, 0] = 0$ and $D[0, j] = 0$.
- If $x[i]$ and $y[j]$ match, i.e., $x[i] = y[j]$, then the LCS is become longer than the previous sequences by one, i.e., $D[i, j] = D[i-1, j-1] + 1$.

- If $x[i]$ and $y[j]$ do not match, i.e., $x[i] \neq y[j]$, then the LCS is the maximum of the previous sequences, i.e., $\max(D[i-1, j], D[i, j-1])$.

The value of $D[m, n]$ is the LCS of the sequences $X = (x[1], \dots, x[m])$ and $Y = (y[1], \dots, y[n])$. The actual LCS sequence can be extracted by traversing the matrix $D[i, j]$.

nLCS algorithm: The nLCS algorithm is derived from the LCS algorithm by replacing the match condition $x[i] = y[j]$ with $(x[i] - y[j]) \leq \text{diff}$ where diff is a tolerance given by a user.

D. nLCSm: LCS for Subsequences with Multi Numerical Attributes

The idea of deriving the nLCS from the LCS can be further extend to the multi numerical attributes to obtain the nLCS for subsequences with multi numerical attributes (nLCSm).

nLCSm algorithm: Let p ($1 \leq p$) denote the number of numerical attributes. Let C_q ($1 \leq q \leq p$) denote the match conditions for the q^{th} numerical attribute. The nLCS is extended with respect to the multiple numerical attributes, named the nLCSm. The nLCSm is derived by replacing the match condition of the nLCS, i.e., $(x[i] - y[j]) \leq \text{diff}$, with $(C_1 \wedge \dots \wedge C_q \wedge \dots \wedge C_p)$.

Figure 7 shows an overview of implementation of the nLCSm algorithm. In principle, the nLCSm algorithm can be implemented by replacing the matching condition of the LCS algorithm with $C_1 \wedge \dots \wedge C_p$ as shown in line 6 of Figure 7. However, the effort to implement the matching condition $C_1 \wedge \dots \wedge C_p$ depends on the complexity of each matching condition.

```

1 for (int i = 0; i <= m; i++) { D[i][0] = 0; }
2 for (int j = 0; j <= n; j++) { D[0][j] = 0; }
3 // == Compute nLCSm
4 for (int i = 1; i <= m; i++) {
5     for (int j = 1; j <= n; j++) {
6         if (C1 ∧ ... ∧ Cp) {
7             D[i][j] = D[i-1][j-1] + 1;
8         } else {
9             D[i][j] = Max(
10                 D[i][j-1], D[i-1][j]);
11         }
12     }
13 }
```

Figure 7. Overview of implementation of the nLCSm algorithm.

In this study, it takes approximately 600 code lines to implement the proposed model consisting of the six parameters shown in Figure 4, which is described in the next section.

E. nLCSm and candlestick pattern retrieval

Given the candlestick pattern model with six parameters as depicted in Figure 4, the nLCSm algorithm can be applied to implementing the model by assigning match conditions C_1 to C_6 for each candlestick as follows.

- C₁: if a difference between closing price change of a reference candlestick and that of a test candlestick is within the change tolerance (*change_tol*), then C₁ is true.
- C₂: if a difference between body length of a reference candlestick and that of a test candlestick is within the body tolerance (*body_tol*), then C₂ is true.
- C₃: if a difference between a closing price of a reference candlestick and a 5-day moving average is within the tolerance (*av5diff_tol*), and a difference of a test candlestick and a 5-day moving average is within the tolerance, then C₃ is true.
- C₄: if a difference between a closing price of a reference candlestick and a 25-day moving average is within the tolerance (*av25diff_tol*), and a difference of a test candlestick and a 25-day moving average is within the tolerance, then C₄ is true.
- C₅: if a difference between a slope of a 5-day moving average of a reference candlestick and that of a test candlestick is within the given tolerance (*slope5_tol*), then C₅ is true.
- C₆: if a difference between a slope of a 25-day moving average of a reference candlestick and that of a test candlestick is within the given tolerance (*slope25_tol*), then C₆ is true.

F. Statistics of candlesticks

In order to estimate the six parameters of the proposed model shown in Figure 4, we calculate the statistics of the candlesticks of the daily NASDAQ composite index from Jan. 2, 2009 to Aug. 20, 2018 of 2,425 business dates.

Figure 8 shows the frequency diagram of body length ratio. Let *Open[i]* and *Close[i]* denote the opening and closing price during the date *i* ($0 \leq i < 2,425$), where *i*=0 means the current day. The body length ratio is computed by the following formula:

$$\text{Body length ratio} = (\text{Close}[i] - \text{Open}[i]) * 100 / \text{Close}[i] \quad (1)$$

The body length ratios are aggregated for every 0.1%. We see that the frequency diagram apparently follows the normal distribution [4]. The average is 0.03216%, while the standard deviation is 0.9330%.

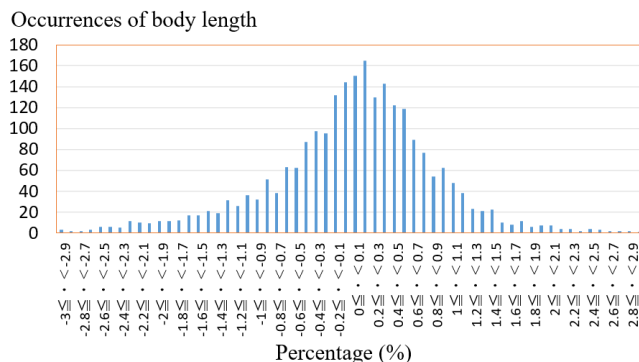


Figure 8. Frequency diagram of body length ratio distributions.

Based on the standard deviation, we divide the body length into seven categories in accordance with the classification of many literatures on candlestick charting [10]. Uptrend and downtrend candlestick bodies are divided into three each. The other one is a *Doji* [4], which is formed when the opening price and the closing price are nearly equal. Table I shows the seven categories of candlestick body. We define the values in Table I so that the candlestick bodies except for the *Doji* occur with the same probability of approximately one sixth. The candlestick categories are involved as a retrieval condition in addition to the tolerances of change, body length, difference from 5-day and 25-day averages, etc.

TABLE I. SEVEN CATEGORIES OF CANDLESTICK BODY.

SDV: Standard deviation of body length ratio.

BD: Parameter for Doji; 0.2% for uptrend, 0.25% for downtrend.

	Upper bound	Lower bound
Long price up body	∞	$0.97 * \text{SDV}$
Medium price up body	$0.97 * \text{SDV}$	$0.44 * \text{SDV}$
Short price up body	$0.44 * \text{SDV}$	BD
Doji	BD	— BD
Short price down body	— BD	$-0.44 * \text{SDV}$
Medium price down body	$-0.44 * \text{SDV}$	$-0.97 * \text{SDV}$
Long price down body	$-0.97 * \text{SDV}$	$-\infty$

Figure 9 shows the frequency diagram of change ratio distributions. Change is defined by the difference between the current value and the previous day's market close. The change ratio is computed by the following formula for each *j* ($0 \leq j < 2,424$):

$$\text{Change ratio} = (\text{Close}[j] - \text{Close}[j+1]) * 100 / \text{Close}[j+1] \quad (2)$$

We see that the frequency diagram is in a form of the normal distribution [4]. The average is 0.07127%, while the standard deviation is 1.149%.

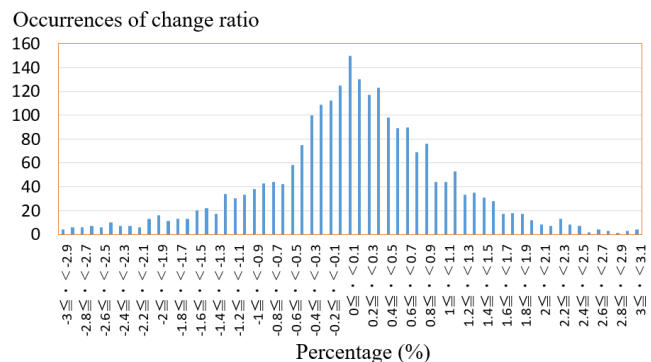


Figure 9. Frequency diagram of change ratio distributions.

Figure 10 shows the frequency diagram of 5-day average ratio distributions. Let *Avg5[k]* denote the 5-day average of the change ratios during the date *k* ($0 \leq k < 2,420$). The 5-day average ratio is computed by the following formula:

$$\text{5-day average ratio} = \frac{(\text{Avg5}[k] - \text{Avg5}[k+1]) * 100}{\text{Avg5}[k+1]} \quad (3)$$

The average is 0.0661%, while the standard deviation is 0.4880%. Since a 5-day average is the mean of five consecutive close prices, the 5-day average ratio statistically follows the standard deviation of the sample [4] of five close prices in theory. In fact, the standard deviation of the 5-day average ratio of 0.4880% roughly equals $1.149\% / \text{SQRT}(5) = 0.5138\%$ with the margin of error of 0.0258%. The error seems to occur because the change ratios do not strictly follow the normal distribution.

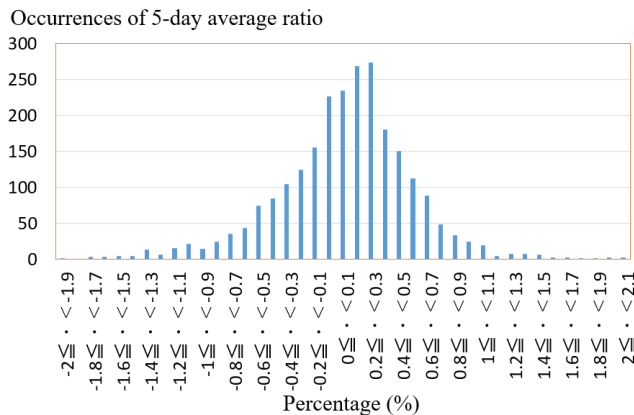


Figure 10. Frequency diagram of 5-day average ratio distributions.

Figure 11 shows the frequency diagram of 25-day average ratio distributions. Let $\text{Avg25}[m]$ denote the 25-day average of the change ratios during the date m ($0 \leq m < 2,400$). The 25-day average ratio is computed by the following formula:

$$\text{25-day average ratio} = \frac{(\text{Avg25}[m] - \text{Avg25}[m+1]) * 100}{\text{Avg25}[m+1]} \quad (4)$$

The average is 0.0680%, while the standard deviation is 0.1910%. The standard deviation of the 25-day average ratio of 0.1910% roughly equals $1.149\% / \text{SQRT}(25) = 0.2298\%$ with the margin of error of 0.0388%.

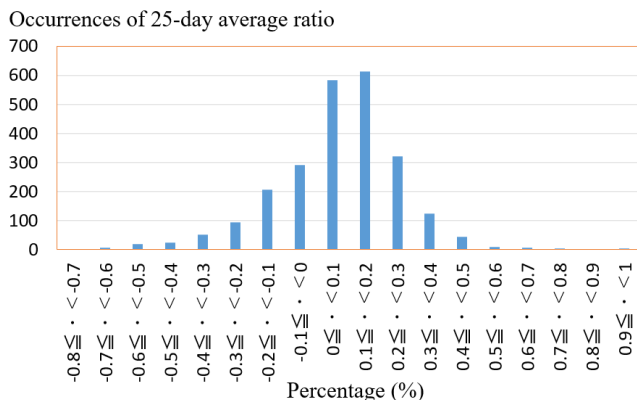


Figure 11. Frequency diagram of 25-day average ratio distributions.

The tolerance of 5-day moving average av5diff_tol is also statistically dependent on the change tolerance change_tol . In the proposed retrieval model, av5diff_tol and av25diff_tol are calculated by the following formulas as defaults according to the definition of the standard deviation of the sample [4].

$$\text{av5diff_tol} = \text{change_tol} / \text{SQRT}(5) = \text{change_tol} / 2.236 \quad (5)$$

$$\text{av25diff_tol} = \text{change_tol} / \text{SQRT}(25) = \text{change_tol} / 5 \quad (6)$$

Therefore, there are essentially four independent parameters in the proposed model, which still causes difficulties in setting parameters. Assuming that each parameter has 5 ranges of values representing, for instance, very high, high, the same level, low, and very low. The candlestick patterns of one candlestick have 5 to the power 4, i.e., $5^4 = 625$ cases of parameters. The patterns composed of two candlesticks have $5^4(4*2) = 625*625 = 390,625$ cases. The patterns of tree candlesticks have $244,140,625$ cases. These cases mean very wide varieties of candlestick charts leading difficulties even in setting parameters for retrieving a specific candlestick chart pattern. In fact, the experiments are performed by repeating trial and error while adjusting parameters.

V. EXPERIMENTAL RESULTS

The predictabilities of the *morning star* pattern indicating a reversal signal of starting uptrend trends, and the *downtrend engulfing* pattern indicating the end of uptrend trends [4] are evaluated through experiments. The experiments are performed on the daily historical stock prices of the NASDAQ composite index of 2,425 business dates from Jan. 2, 2009 to Aug. 20, 2018.

A. Data Conversion

The stock prices are converted to the ratio of closing prices to reduce the effects of highness or lowness of the stock prices. The formula below is used for calculating the ratio of prices in a percentage.

$$\text{RCP}_j = (\text{CP}_j - \text{CP}_{j+1}) * 100 / \text{CP}_{j+1} \quad (1 \leq j \leq n) \quad (7)$$

CP_j indicates the closing price of the j -th business date. CP_1 means the closing price of the current date. RCP_1 is the ratio of the difference between CP_1 and CP_2 , and the closing price of CP_2 , i.e., the closing price of the date before the current date. The similar calculations are performed to opening, high, and low prices. In addition, the 5-day and 25 day moving averages, and their slopes are calculated before the experiments. The number of valid 25 day moving averages, i.e., n in effect is 2,400 ($= 2,425 - 25$) because the 25-day averages can't be calculated to the last 25 days.

B. Experiments on Morning Star Pattern

In the previous contribution [1], we observe that the morning star pattern does not necessarily show stable uptrend signal, and one confirmation day after the pattern significantly improves the predictability of the pattern. This

is true for the NASDAQ market. So we start this section with the morning star pattern plus one confirmation day.

Figure 12 shows the candlestick chart of the NASDAQ in which a strong uptrend starts on May 4, 2018 with a long uptrend body of 1.9966%. The short uptrend body of 0.3219% follows the next day, i.e., May 7, 2018. The candlestick arrangements from May 1 to May 7, 2018 form a morning star pattern with one confirmation day.

The first experiment is performed on the four candlesticks surrounded by a solid rectangle in Figure 12 as a reference for retrieval. This means the length of the reference candlesticks is four. We set the length of the candlesticks to be retrieved to eight, i.e., double the length of the reference.

The change tolerance varies from 0.3% to 1.8%. The tolerances of 5-day and 25-day moving average are calculated by (5) and (6). The slope of 5-day moving average is set to 0.4880% in an upward direction, i.e., the standard deviation of the 5-day average ratio, and $-0.4880 \times 0.25\%$ in a downward direction. The slope conditions mean to select approximately 44% of the original candlesticks. The slope of 25-day moving average is set analogously.

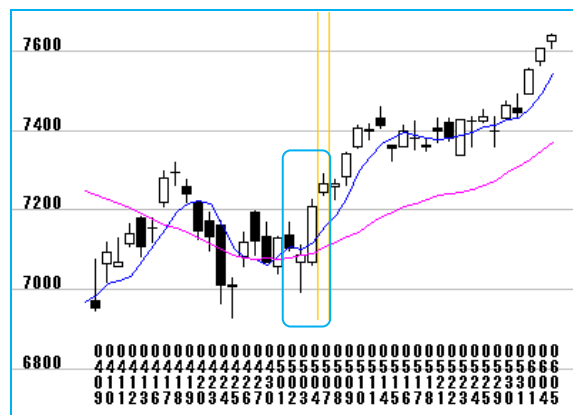


Figure 12. Candlestick chart around May 7, 2018.

Figure 13 shows the number of retrieved business dates that have a candlestick pattern similar to that of May 7, 2018 within a change tolerance ranging between 0.3% and 1.8%. The number of data gradually increases as the change tolerance rise, but it does not rise linearly. The algorithm retrieves almost the same number of business dates for tolerance between 0.8% and 1.2%, and between 1.3% and 1.8%.

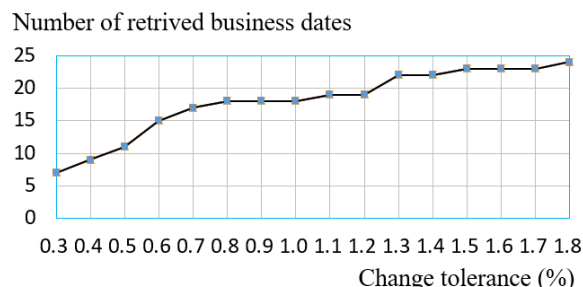


Figure 13. Number of retrieved business dates and change tolerance.

Table II shows a list of the business dates retrieved under the change tolerance of 1.1%. “nLCSm” shows the length of candlesticks that match the key for retrieval. “Length” represents the length of the retrieved candlesticks. “nLCSm/Length” indicates the “density” of the matched candlesticks in the retrieved ones. So, “ $nLCSm^2/Length$ ” means the “weight” of the retrieved candlesticks and used as a key to sort in descending order.

TABLE II. RESULTS CONCERNING MORNING STAR PATTERN PLUS ONE CONFIRMATION DAY ON MAY 7, 2018 UNDER THE CHANGE TOLERANCE OF 1.1%.

Bus.Date	nLCSm	Length	nLCSm/Length	$nLCSm^2/Length$
20180507	4	4	1.000	4.000
20120119	4	5	0.800	3.200
20180709	4	6	0.667	2.667
20160711	4	6	0.667	2.667
20160525	4	6	0.667	2.667
20130503	4	6	0.667	2.667
20150319	4	7	0.571	2.286
20131223	4	7	0.571	2.286
20121129	4	7	0.571	2.286
20110705	4	7	0.571	2.286
20171218	3	4	0.750	2.250
20151016	3	4	0.750	2.250
20130411	3	4	0.750	2.250
20121120	3	4	0.750	2.250
20110527	3	4	0.750	2.250
20100914	3	4	0.750	2.250
20090427	3	4	0.750	2.250
20150123	4	8	0.500	2.000
20100308	4	8	0.500	2.000

Figure 14 shows overlapped closing prices whose business dates are listed in Table II for graphically representing the future stock trend. All business dates are aligned on the origin to make the comparison easy. The thick black line represents the closing prices of the reference date, i.e., May 7, 2018. Thin solid lines represent the closing prices of business dates listed in Table II except for the reference date. The thick light blue line indicates the average of the candlestick charts plotted by thin solid lines, which suggests at most 1% increase in the coming 10 days.

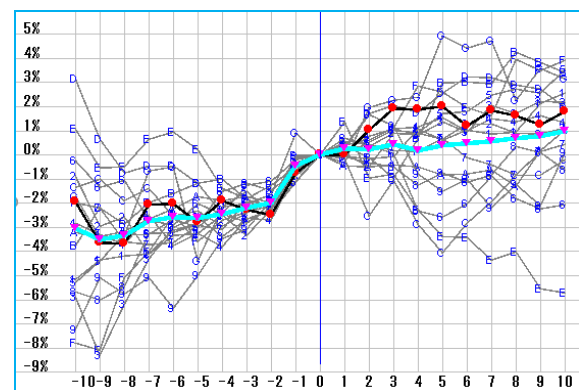
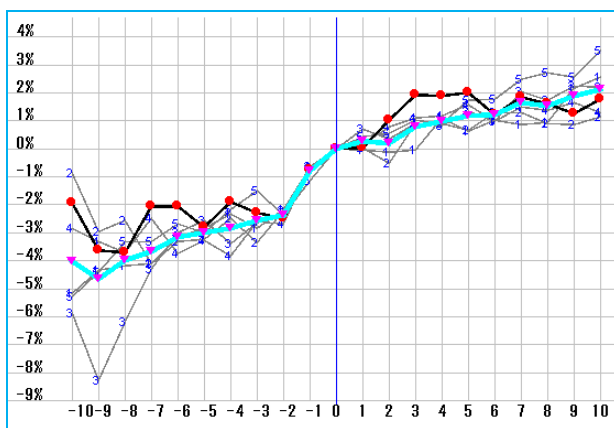


Figure 14. Overlapped closing prices of retrieved candlesticks using May 7, 2018 as key.

price movements. It is vital to control the number of retrieved data to be appropriate for prediction.

Figure 1 is a line graph showing the evolution of the average of the normalized difference between the estimated and true functions, $E[|\hat{f} - f|]$, over 10 iterations. The x-axis represents iterations from -10 to 10, with a vertical blue line at iteration 0. The y-axis represents the normalized difference, ranging from -9% to 4%. The graph displays multiple black lines with markers, a thick cyan line, and a thick black line. The values generally increase from negative to positive after iteration 0.

Figure 16 shows overlapped closing prices of business dates whose weight are not less than 2.667. All five closing prices suggest an uptrend. This result may well be noteworthy for traders to identify buying opportunities with expectation of approximately 2.0% profits on average in the next ten days.



We should learn from the retrieval results shown in Figure 14 through 16 that there are price movements opposite to what candlestick pattern predicts. As the number of the retrieved candlestick sequences increases, the average value of the retrieved sequences tends to approach 0, meaning that candlestick pattern technique has no predictive power on

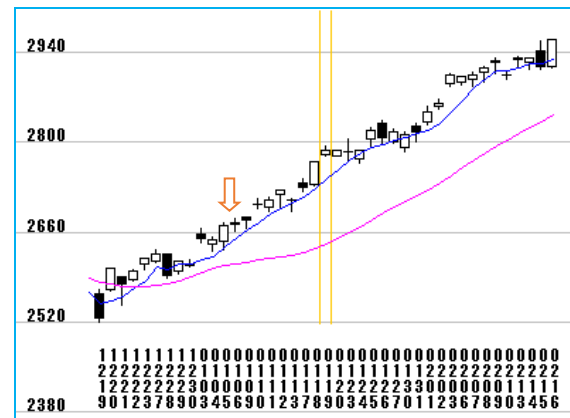


Figure 18 shows the candlestick chart around July 11, 2016. This chart is similar to that of May 7 in Figure 12 in the sense that a few days passed after the 5-day average exceeds the 25-day average. The chart is worthy of the investor to pay attention to.

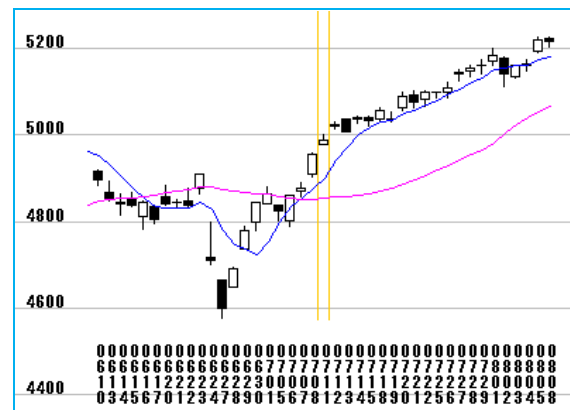


Figure 19 shows the candlestick chart around July 9, 2018. The key candlestick pattern of retrieval just comes up on the day that the 5-day average exceeds the 25-day average, which seems to imply a rather weak uptrend. In fact, the opening and closing prices move in a zigzag way, and draw a downtrend after 14 days. The candlestick chart around May 25, 2016 in Figure 20 shows the same feature and changes the trend after 10 days.

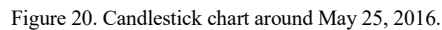
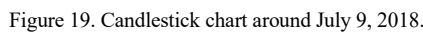


Figure 21 shows the candlestick chart of the NASDAQ around June 21, 2018 with a long downtrend body preceded by a short candlestick. The pattern is named the *downtrend engulfing* pattern [4] suggesting downtrend reversal. The second experiment is performed on the four candlesticks surrounded by a solid rectangle in Figure 21 as a reference for retrieval. We set the length of the retrieved candlesticks to eight, which is the same as the first experiment.

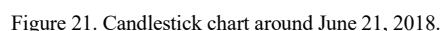


Figure 23 shows overlapped closing prices whose business dates are listed in Table III. The thick light blue line, representing the average of the retrieved candlesticks, indicates that the sideways are expected after two-days short dips with the wide price fluctuations ranging between 3.5% and -8.9% .

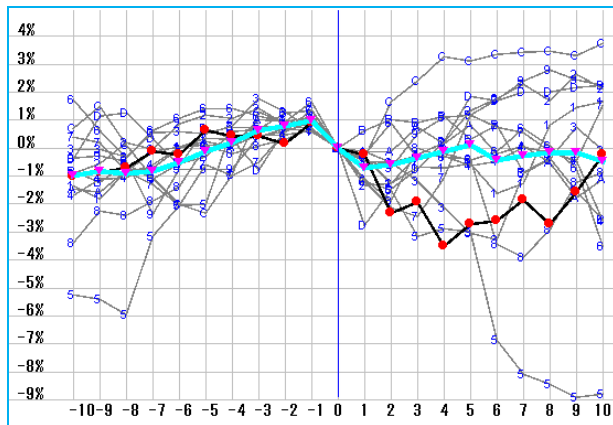


Figure 23. Overlapped closing prices of retrieved candlesticks using June 21, 2018 as key.

Figure 24 shows overlapped closing prices of business dates whose weight are not less than 2.0. Five business dates with the weight between 1.286 and 1.80 are discarded including one forming the upper bound. As a result, the average closing prices of the retrieved candlesticks decreases gradually to 1%.

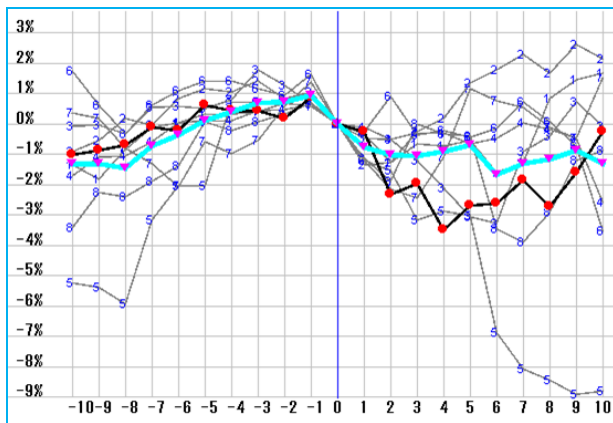


Figure 24. Overlapped closing prices of business dates whose weight are not less than 2.0.

Figure 25 shows overlapped closing prices of business dates whose weight are not less than 2.25. The prices of the business dates with the weight between 1.286 and 1.80 are located between the upper bound and the lower bound. Consequently, the average closing prices of the retrieved candlesticks seems to remain the same as that of Figure 24.

Figure 25 suggests two possible price trends after two-day drips. One is an uptrend reversal and the other is a continuous downtrend trend as shown by the upper and lower bounds of Figure 25, i.e., lines marked by “2” and “5”.

Figure 26 shows the candlesticks around Nov. 30, 2016, which forms the upper bound depicted in Figure 25. Figure 27 shows the candlesticks around June 21, 2010 forming the lower bound.

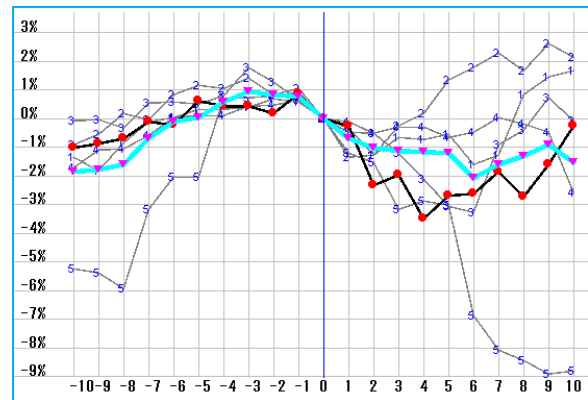


Figure 25. Overlapped closing prices of business dates whose weight are not less than 2.25.

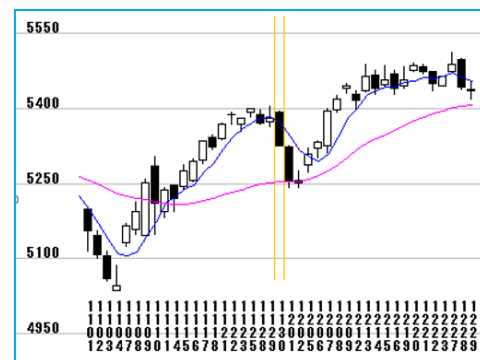


Figure 26. Candlestick chart around Nov. 30, 2016.



Figure 27. Candlestick chart around June 21, 2010.

The candlestick chart in Figure 26 shows the downtrend engulfing pattern on Nov. 30, 2016, while forms the *morning star* pattern three days later, i.e., Dec. 5.

Figure 27 shows a continuous downtrend. To the best our knowledge, the candlestick charting can tell when the specified trend begins, but *cannot* tell when the reverse of the specified trend begins. The results suggest the limitation of the stock price prediction by the candlestick charting.

In the study, we find that the prediction of the downtrend trend is more difficult than that of the uptrend trend. The fact seems to reflect that the NASDAQ stock market is constantly

growing. In fact, the NASDAQ index on Aug. 20 2018 is 2.16 times the price on Aug. 20 2013.

VI. CONCLUSION AND FUTURE WORK

Extracting stock price change patterns and predicting future stock prices are an interesting task for traders as well as financial data analysts. Typically, when there is no outstanding news, stock price is apt to show small price change during the period. These small price changes are generally interpreted as market indecisiveness and need to be eliminated when retrieving similar stock prices.

In this paper, we propose an algorithm for matching candlesticks while skipping small price changes. A numerical sequence version of the LCS (Longest Common Substring) algorithm, which makes use of dynamic programming technique, is devised for the purpose.

This paper also proposes a model with six parameters that provide bases for retrieving similar candlestick patterns. The proposed model is original because it deals with slopes of the 5-day and 25-day moving averages to identify their trends, in addition to 5-day and 25-day moving averages to decide whether the price occurs in high or low price zones.

In this study, we employ statistical values of the candlesticks to estimate the parameters of the model. The standard deviation of the sample [4] is substantially helpful to approximate the parameters concerning the 5-day and 25-day moving averages.

Experiments are performed on the daily NASDAQ composite index. The results of the experiments seem to show that the forecast for the uptrend trend is more accurate than that of downtrend, which reflects that the NASDAQ stock market is constantly growing.

We also find limitations of the current approach though the experiments. First, the determination of the value of the parameters is still immature. Actually, it is done by trial and error all through the experiments. We need to develop a systematic method to estimate the value of the parameters and the number of retrieved candlestick data. Second, the proposed model fails to support trading volume. According to literature on technical analysis [4], the volume is an important indicator to confirm the strength or weakness of price movements. Finally, the proposed model need to support parameters on upper and lower shadows. Support of shadow parameters allows the model to examine various shadow sensitive patterns known as the *hammer*, *dragon fly* patterns [4].

As the current study is limited to an average or composite index of stock markets, future researches will focus on analyzing stock prices of industry sectors and individual companies in international stock markets.

Our future plans also include developing an automatic pattern mining method to discover frequently repeated patterns for stock trend prediction. Since candlestick patterns proposed so far have been found from human experience, there can be unknown profitable candlestick patterns with more complicated structures than ever known. Finding noticeably profitable patterns stimulates our interest in researches of data mining.

ACKNOWLEDGMENT

This research is supported by the JSPS KAKENHI under grant number 16K00161.

REFERENCES

- [1] Y. Udagawa, "Design and Implementation of Candlestick Chart Retrieval Algorithm for Predicting Stock Price Trend," The Fourth International Conference on Big Data, Small Data, Linked Data and Open Data (ALLDATA 2018), pp. 19–25, 2018.
- [2] Yahoo! Finance, "NASDAQ Composite, Historical Data," <https://finance.yahoo.com/quote/%5EIXIC/history?p=%5EIXIC>, Aug. 2018.
- [3] V. Drakopoulou, "A Review of Fundamental and Technical Stock Analysis Techniques," Journal of Stock & Forex Trading vol. 5, pp. 1–8, Nov. 2015.
- [4] "Technical Analysis," Cambridge Univ. pp. 1–179, Available from: <http://www.mrao.cam.ac.uk/~mph/TechnicalAnalysis.pdf>, Feb. 2011.
- [5] Fusion Media Limited. "Nikkei 225 Futures," <https://www.investing.com/indices/japan-225-futures-candlestick>, Aug. 2018.
- [6] J. M. Horton, "Stars, crows, and doji: The use of candlesticks in stock selection," Quarterly Review of Economics and Finance, vol. 49, pp. 283–294, Nov. 2007.
- [7] R. B. Marshall, R. M. Young, and R. Cahan, "Are candlestick technical trading strategies profitable in the Japanese equity market?," Review of Quantitative Finance and Accounting, vol. 31, pp. 191–207, Aug. 2008.
- [8] P. Tharavanij, V. Siraprasiri, and K. Rajchamaha, "Profitability of Candlestick Charting Patterns in the Stock Exchange of Thailand," SAGE journals, pp. 1–18, Oct. 2017.
- [9] G. Caginalp and H. Laurent, "The predictive power of price patterns," Applied Mathematical Finance, vol. 5, 1998, pp. 181–206.
- [10] Y.-J. Goo, D.-H. Chen, and Y.-W. Chang, "The application of Japanese candlestick trading strategies in Taiwan," Investment Management and Financial Innovations, vol. 4, pp. 49–79, Jan. 2007.
- [11] C. Chootong and O. Sornil, "Trading Signal Generation Using A Combination of Chart Patterns and Indicators," International Journal of Computer Science Issues, vol. 9, pp. 202–209, Nov. 2012.
- [12] T.-H. Lu, Y.-C. Chen, and Y.-C. Hsu, "Trend definition or holding strategy: What determines the profitability of candlestick charting?," Journal of Banking & Finance, vol. 61, Dec. 2015, pp. 172–183.
- [13] M. Zhu, S. Atri, and E. Yegen, "Are candlestick trading strategies effective in certain stocks with distinct features?," Pacific Basin Finance Journal, vol. 37, pp. 116–127, Apr. 2016.
- [14] F. Martinsson and I. Liljeqvist, "Short-Term Stock Market Prediction Based on Candlestick Pattern Analysis," Examensarbete Inom Teknik, Stockholm, Sverige, June, 2017, pp. 1–36.
- [15] C.-L. Chin, M. Jais, S. S. Balia, and M. Tinggi, "Is candlestick continuation patterns applicable in Malaysian stock market?," SHS Web of Conferences 34, 2017.
- [16] "Longest common subsequence problem," https://en.wikipedia.org/wiki/Longest_common_subsequence_problem, Sept. 2018.
- [17] E. F. Fama, "Efficient capital markets: A review of theory and empirical work," Journal of Finance, vol. 25, 1970, pp. 383–417.

POMVCC: Partial Order Multi-Version Concurrency Control

Yuya Isoda, Atsushi Tomoda, Tsuyoshi Tanaka, Kazuhiko Mogi

Hitachi, Ltd. Research & Development Group

1-280, Higashi-koigakubo, Kokubunji-shi, Tokyo, Japan

email: { yuuya.isoda.sj, atsushi.tomoda.nx, tsuyoshi.tanaka.vz, kazuhiko.mogi.uv } @ hitachi.com

Abstract — This paper proposes Partial Order Multi-Version Concurrency Control (POMVCC), which is a concurrency control technique based on partial order transaction processing. We claim that timestamp generation per transaction can be a critical section on multi-core for high-throughput DataBase Management Systems (DBMSs), and POMVCC can execute multiple transactions using the same timestamp without losing consistency. In this paper, we change the order of transaction processing from total to partial on Multi-Version Concurrency Control (MVCC), which allocates a timestamp on partial order per multiple transactions. It helps the DBMS reduce the overall number of increments to the timestamp; therefore, improving its overall performance. We claim that a POMVCC-based system achieves 1.74 times higher throughput than that of a conventional MVCC-based system. We implemented a lock-free version of POMVCC on MPDB, which is in-memory DBMS.

Keywords – Partial order transaction processing; Multi-version concurrency control; Transaction; Timestamp; In-memory DB.

I. INTRODUCTION

We research to adapt new hardware technology or new software techniques to old DataBase Management Systems (DBMS) techniques [1][2][3]. For example, the number of Central Processing Unit (CPU) cores and memory size have recently increased due to the progress of hardware technology. For DBMSs, scalability technology [4][5][6] for multicore CPUs and large-scale and non-volatile in-memory technology [7][8] are advancing rapidly, and the performance of DBMSs is close to reaching one million Transactions Per Second (tps) [5][9].

A DBMS must guarantee the Atomicity, Consistency, Isolation and Durability (ACID) properties to maintain data consistency [10]. However, strictly doing so prevents a DBMS from improving performance because it needs to process Transactions (Tx) as serial processing in total order [11]. To improve performance, a DBMS generally uses the isolation level, which lessens ACID properties step by step; thus, improving parallel processing.

Multi-Version Concurrency Control (MVCC) has recently been used for controlling the isolation level. It manages timestamps of both before and after updating a record and enables records to be referenced and updated simultaneously. As a result, it increases the performance of OnLine Transaction Processing (OLTP). Recent research has also clarified how SERIALIZABLE can be implemented. Therefore, DBMSs with MVCC are expected to become widespread in the near future [12][13].

There are two types of Timestamps (Ts) for MVCC, i.e., physical clock and logical clock. The physical clock is the time used in the real world, such as Coordinated Universal Time (UTC). The Network Time Protocol (NTP) is widely used as a protocol for synchronizing UTC among servers [14]. Implementation of a logical clock in DBMSs is common [15]. Spanner implemented a physical clock for DBMSs, but such an example is rare [16]. The larger the system is, the more difficult conventional timestamp management becomes using a logical clock. Because it is mandatory for timestamps to be numbered every 1 us to reach one million tps. In such an environment, large-scale mutual exclusion with a high CPU clock frequency may be problematic. In addition, the memory size and the number of CPU cores will increase, e.g., Hewlett Packard's Memory-Driven Computing, will further increase [17].

Silo was proposed to solve this problem [9]. Silo is the timestamp based on Epoch. It periodically updates the high-order bits of the timestamp. Transaction threads update low-order bits under the condition that they satisfy the order of dependence. As a result, Silo can reduce the number of updates for the timestamp. However, it cannot be easily adapted for conventional MVCC-based DBMSs because it requires lock processing and management of the Read-Set and Write-Set for concurrency control.

Moreover, we must better understand the partial order model and low isolation levels because a user requires two advanced points. The first point is high-performance and high-scalability. NoSQL is very fast and executes 80–120 million operations per second [18]. If we want to promote only DBMS to a data management system for simple management, the performance of DBMS needs to exceed the one of NoSQL. The second point is that we must understand the meaningless assumptions on industry, as shown in Figure 1 [19]. High isolation levels and the stored procedures are not needed on industry. Not all transactions are executed as stored procedures only 47% of users (excluded 0% and 1–10% on Figure 1.B), and almost all users do not set the isolation level of SERIALIZABLE. Read Committed is most frequently used; therefore, we need to develop a high-performance DBMS on a low isolation level.

From these reasons, we propose Partial Order Multi-Version Concurrency Control (POMVCC), which is the partial order transaction processing based on the reduction in the conflict rate, which is caused by a large-scale DB. It mitigates the problems with simultaneous executable transactions on each isolation level. Specifically, it increments a timestamp during the abortion phase of a transaction. Thus, multiple transactions can be processed at

the same timestamp, and the number of timestamp updates can be reduced on any isolation levels.

In summary, our contributions are as follows.

1. We propose partial order transaction control based on reconsidering the isolation level of MVCC, called POMVCC. To update a timestamp during the abortion phase of a transaction, POMVCC can process multiple transactions at the same timestamp and reduce the number of timestamp updates. It is also easily implementable for DBMSs based on MVCC.
2. We show the cause and solution of a new anomaly called “HISTORICAL READ” caused by POMVCC.
3. We also show a lock-free implementation of POMVCC and discuss the implementation of mixed Pessimistic Concurrency Control (PCC) and Optimistic Concurrency control (OCC) to solve the problem of long-short transaction.
4. Finally, we discuss the implementation of POMVCC on an in-memory DBMS and the evaluation its performance.

The rest of this paper is organized as follows. In Section II, we introduce research on concurrency control for DBMSs. In Section III, we reconsider the requirement of concurrency control for DBMSs and present a problem with performance and scalability. In Section IV, we propose POMVCC and discuss a new anomaly called “HISTORICAL READ” caused by POMVCC and its solution. In Section V, we describe a method for implementing our developed MPDB, which is an MVCC-based, lock-free, in-memory DBMS characterized by parallel logs and mixed PCC/OCC. In Section VI, we describe a method for implementing POMVCC that is lock-free. In Section VII, we discuss the evaluation of POMVCC’s performance and present the results. Finally, in Section VIII, we give concluding remarks and discuss our future work.

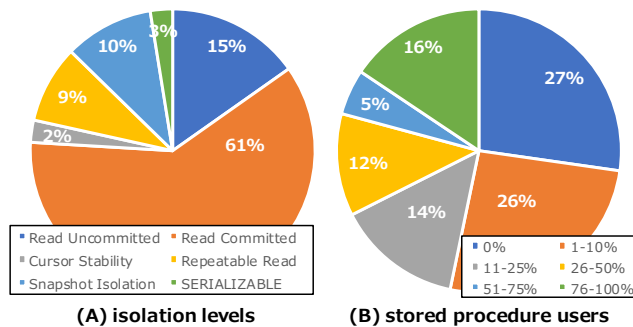


Figure 1. Survey on frequency of use on DBMS functions

II. RELATED WORK

In this section, we discuss work related to concurrency control for DBMSs. The most notable viewpoint of concurrency control is the durability of an execution result and the concurrency control of transactions.

Algorithms for Recovery and Isolation Exploiting Semantics (ARIES) involve general persistence processing [20]. ARIES is composed of analysis, REDO, and UNDO.

Analysis pinpoints the starting point of a recovery, REDO re-executes a transaction on the basis of a REDO log, and UNDO deletes an uncommitted transaction on the basis of an UNDO log. During logging, Write-Ahead Logging (WAL), which can restore logs safely in the case of failure, is used. WAL has a problem in that the speed of writing a log to a storage device is slow. However, faster technology that uses distributed logging with non-volatile memory has recently been proposed for WAL [7].

Research on the concurrency control of transactions has been conducted since the 1980s. There are two types of concurrency control, i.e., PCC and OCC [21][22][23]. For PCC, concurrency control with a 2-Phase Lock (2PL) is mainly used. DORA [24], PLP [25], and Shore-MT [26] have been proposed as lock-based DBMSs [27]. However, DBMSs with MVCC, which enables OCC, have recently been proposed because the processing cost of locks and latches is high [28][29][30].

It was stated that an isolation level for SERIALIZABLE is not possible [31]. However, the proposal of SERIALIZABLE SNAPSHOT ISOLATION (SSI) has made this possible [12][13]. Using this technology, H-Store/VoltDB [32][33], Hekaton [4][6], and SAP HANA [8] were proposed as MVCC-based DBMSs. H-Store creates transaction sites, the number of which is the same as the number of CPUs, and transaction threads that stick to the logical sites execute Structured Query Language (SQL). Such a mechanism enables in-memory and lock-free fast processing. To reduce the number of responses between interfaces, Hekaton compiles stored procedures into native codes. SAP HANA manages both the row store, the update efficiency of which is high, and column store, the reference efficiency of which is high. Many such MVCC-based DBMSs that have diverse characteristics have been proposed.

Moreover, a Silo in-memory DBMS that manages Epoch-based timestamps as a concurrency control has also been proposed [9]. In Silo, updates of timestamps are removed from the concurrency control of a transaction on Single-Version Consistency Control (SVCC). Silo uses a special-purpose thread for managing timestamps. As a result, it achieves high-performance. In addition, it creates temporary areas per transaction for references (Read-Set) and updates (Write-Set). Concurrency control with Read-Set and Write-Set can use cache and memory efficiently. Using these technologies, Silo achieves 700,000 tps for the industry standard benchmark TPC Benchmark™ C (TPC-C) [34]. Moreover, Silo-based transaction control is adopted by Intel’s Rack-Scale Architecture, which has become popular, and in-memory DBMS Foedus [5], which uses Hewlett Packard’s Memory-Driven Computing [17]. Therefore, Silo-based concurrency control has become popular.

Research on SVCC-based DBMSs is now advancing. Silo-like concurrency control enables faster than conventional MVCC-based DBMSs. However, it is difficult to adopt it for MVCC-based DBMSs because many components, such as thread management, transaction control, and data management, must be modified. Therefore, we propose an easier implementation technique that is equivalent to Silo’s concurrency control for MVCC-based DBMSs.

III. RECONSIDERING ANOMALIES AND CONCURRENCY CONTROL ON MVCC

In this section, we outline concurrency control on MVCC and reconsider the update conflict of timestamps, which is a problem in Silo, and solve this problem.

A DBMS must maintain ACID properties, but to do so strictly, transactions must be serialized, which degrades performance. To avoid this phenomenon, an isolation level, in which ACID properties are lessened gradually, is used. The isolation level is defined as the allowable range for an anomaly, which occurs when transactions are executed in parallel. This mitigation achieves high-scalability enabled by the highly parallel and high-performance transactions of DBMSs.

The isolation level differs between lock-based control and MVCC-based control [31]. We outline the relationship of the isolation level for MVCC and anomalies and clarify the order of transactions and mitigate the problem with scalability.

We define B as the begin phase of a transaction, C as the commit phase of the transaction, A as the abort of the transaction, BTs as an allocated timestamp during the begin phase, CTs as an allocated timestamp during the commit phase, ATs as an allocated timestamp during the abort phase, R as the reference in the transaction, and W as the update/insert/delete in the transaction. We also define Tx.1, Tx.2, etc., as identifiers of transactions X, Y, etc. as a set of records and i, j, etc. as integers.

A. Relationship between isolation level and anomalies

The general anomalies are WRITE SKEW (WS), FUZZY READ (FR), READ SKEW (RS), and LOST UPDATE (LU) on MVCC [31]. Examples of these anomalies are listed in Table I.

For example, LOST UPDATE occurs when Tx.1 and Tx.2 update record X simultaneously and both are successful. This is a problem because the value of the record is either X' or X'', and the update history of the record is not uniquely determined. For one-side failure (W1 W2 C2 A1), LOST UPDATE may occur when Tx.2 updates record X to X', then Tx.1 aborts and record X' is roll-backed to X.

The isolation level is defined as the allowable range for anomalies. SSI has the strictest requirement of consistency. The second strictest is READ COMMITTED and READ UNCOMMITTED is the least strict. Table II lists the relationships between the isolation level and anomalies. For example, for READ COMMITTED, WRITE SKEW or FUZZY READ may occur. READ UNCOMMITTED is hardly used because user-unallowable anomalies occur.

TABLE I. ANOMALIES ON MVCC

Anomaly	Formula
LOST UPDATE (LU)	$W2[X \rightarrow X'] \ W1[X \rightarrow X'']$
READ SKEW (RS)	$W2[X \rightarrow X', Y \rightarrow Y'] \ R1[X', Y]$
FUZZY READ (FR)	$R1[X] \ W2[X \rightarrow X'] \ R1[X']$
WRITE SKEW (WS)	$R1[X] \ R2[Y] \ W1[Y \rightarrow Y'] \ W2[X \rightarrow X']$

TABLE II. ISOLATION LEVELS ON MVCC

Isolation Level	LU	RS	FR	WS
SERIALIZABLE	-	-	-	-
SNAPSHOT ISOLATION	-	-	-	v
READ COMMITTED	-	-	v	v
READ UNCOMMITTED	v	v	v	v

B. Concurrency control

MVCC controls records and transactions by using a timestamp. MVCC manages the update history of records by allocating a timestamp at the commit to the records. Transactions refer to a timestamp at the begin phase or when SQL executes and to the latest record whose timestamp is smaller than BTs. The references of transactions maintain consistency with this method. How BTs is treated differs depending on the isolation level. SERIALIZABLE and SNAPSHOT ISOLATION use a timestamp that is referred to at the begin phase. READ COMMITTED uses a timestamp that is referred to at SQL execution. Figure 2 shows the difference between Tx.2 as SNAPSHOT ISOLATION and Tx.3 as READ COMMITTED. They execute the SQL at the same time. However, Tx.2.SQL2 reads X, but Tx.3.SQL2 reads X'. Such concurrency control protects SNAPSHOT ISOLATION from FUZZY READ. Similarly, READ SKEW is prevented.

The update conflicts at the validation of the commit process generally use First Committer Win (FCW), which is an OCC. It executes transactions in the order in which the commit command is executed. It maintains consistency by aborting subsequent conflicting transactions.

The concurrency control explained above cannot prevent WRITE SKEW from occurring. This occurs when references and updates of multiple transactions mutually conflict (RW-Conflict). SSI was proposed to find such a condition and avoid WRITE SKEW [12][13]. SSI adds a read flag and write flag to the conventional MVCC algorithm and detects RW-Conflict. It aborts at least one of the RW-Conflict transactions and avoids WRITE SKEW. Therefore, SERIALIZABLE is enabled. SSI enables SERIALIZABLE with the same performance of SNAPSHOT ISOLATION [12][13]. Thus, we can prevent anomalies from occurring by using these concurrency controls on MVCC.

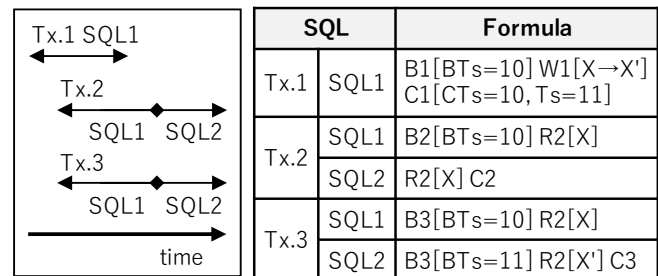


Figure 2. Difference between SNAPSHOT ISOLATION (Tx.2) and READ COMMITTED (Tx.3)

C. Problem of scalability

To strictly maintain ACID properties, it is necessary for transactions to be processed in total order. Scalability is low in this case. Table III defines D1 as total order, D2 as weak order, and D3 as the order of transactions for MVCC.

The CTs of MVCC must be different between the allocation times of the transactions; one of the transactions must be the reference transaction. That is, multiple update transactions cannot be committed at the same time due to D3. Thus, the transactions of MVCC are in total order in the case of update transactions only, or it is in weak order when transactions include reference transactions.

As described above, MVCC increases scalability; however, it is applicable only for transactions including reference transactions. In the case of update transactions only, scalability is low because the conditions of the order are the same as D1. Therefore, we must mitigate the order of update transactions under D3. ii in order to improve the scalability for DBMS.

TABLE III. DEFINITION OF MVCC

D1. Total Order	
$i < j \iff i \leq j \text{ and } i \neq j$	
D2. Weak Order	
$i \leq j \iff i < j \text{ or } i = j$	
D3. MVCC for write tx.	
CTs(Tx.i) < CTs(Tx.j) \iff i and ii	
i	CTs(Tx.i) \leq CTs(Tx.j)
ii	CTs(Tx.i) \neq CTs(Tx.j)

IV. PROPOSAL OF POMVCC

In this section, we propose POMVCC, which mitigates the order of update transactions and enables high-scalability. We also consider a new anomaly caused by POMVCC.

We define DBC as the content of a database, and the execution order of transactions is shown as (\rightarrow).

A. Basic idea

Transactions can be controlled in partial order on the basis of the consistency of a DBC. For example, if the concurrency control of DBMS exchanges the execution order of one transaction with another transaction and the result does not change, these transactions can be executed in non-order, and consistency is maintained. Thus, we do not need to update the timestamp per transaction update and can share one timestamp among multiple update transactions. Therefore, we propose POMVCC as a new concurrency control focused on the partial order of transactions. POMVCC provides the same timestamp to two update transactions if they have no dependency. This technique mitigates condition D3. ii, so scalability can increase.

The concept and definition of POMVCC are shown in Figure 3 and Table IV. By controlling the partial order of transaction processing, POMVCC eliminates the need to update the timestamp every time transaction process is ended. POMVCC updates the timestamp when it detects an anomaly. For example, in Figure 3, since LOST UPDATE occurred between Tx.1 and Tx.3, POMVCC will update the timestamp. Even if the execution order of all transaction processes within the same timestamp is changed, POMVCC permits simultaneous execution if the content of the database is the same.

We show the allowable conditions of transaction processing on the same timestamp for MVCC and POMVCC in Table V, which shows that POMVCC has more conditions that can be executed simultaneously than MVCC. Therefore, POMVCC can reduce the update frequency of timestamps. This means that the scalability of POMVCC is better than that of MVCC. We discuss the difference in isolation levels between MVCC and POMVCC, as shown in Figure 4.

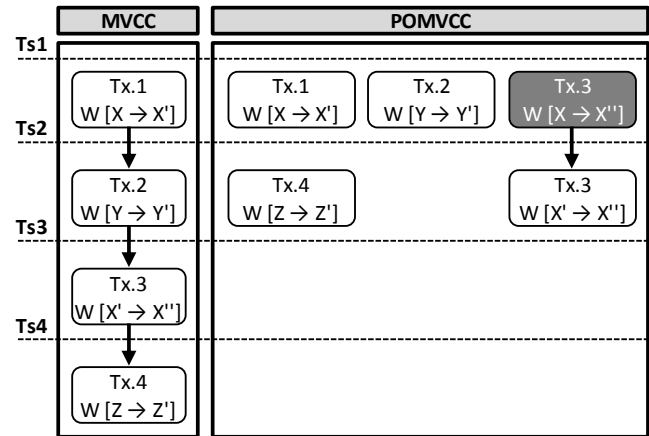


Figure 3. Difference between MVCC and POMVCC

TABLE IV. DEFINITION OF POMVCC

D4. POMVCC for write tx.	
CTs(Tx.i) \leq CTs(Tx.j) \iff I or II	
I	CTs(Tx.i) < CTs(Tx.j)
II	CTs(Tx.i) = CTs(Tx.j) and DBC(Tx.i \rightarrow Tx.j) = DBC(Tx.j \rightarrow Tx.i)

TABLE V. ALLOWABLE CONDITIONS OF TRANSACTION PROCESSING ON SAME TIMESTAMP FOR MVCC AND POMVCC

Formula	MVCC	POMVCC
R1[X] R2[X]	Success	Success
R1[X] W2[X]	Success	Success
W1[X] R2[X]	Success	Success
W1[X] W2[Y]	Failure	Success
W1[X] W2[X]	Failure	Failure

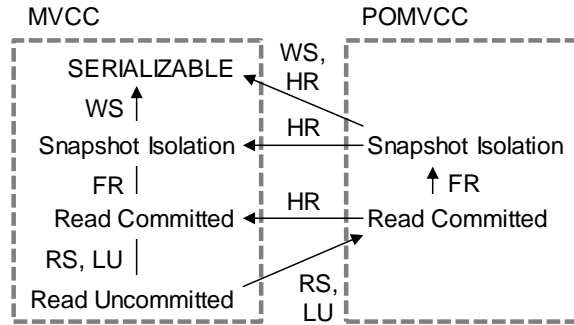


Figure 4. Diagram of isolation levels and relationships

B. How to control POMVCC

The trigger to update a timestamp of POMVCC differs from that of MVCC. MVCC updates a timestamp during the commit phase of a transaction, but POMVCC updates it during the abort phase of a transaction. Thus, multiple update transactions can be executed at the same timestamp on POMVCC.

The protocol of POMVCC is shown in Figure 5. The conflict of LOST UPDATE occurs between Tx.1 and Tx.3 on record X. In the case of MVCC, a timestamp is updated at the commit of Tx.1, but in the case of POMVCC, a timestamp is not updated. Therefore, Tx.3 refers to old record X, and conflict occurs. POMVCC updates a timestamp at the abort of Tx.3. Record X can be updated when Tx.3 is retried. Because a timestamp is updated at the abort of a transaction caused by an anomaly, partial order transaction control is possible.

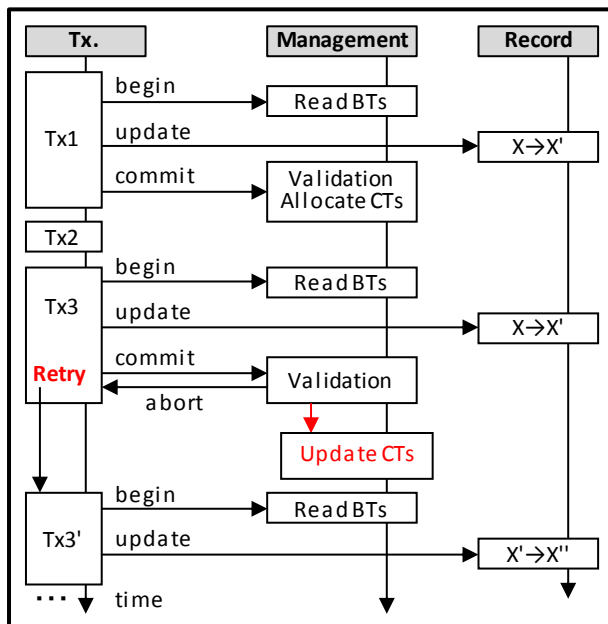


Figure 5. Concurrency control of POMVCC

C. New anomaly: HISTORICAL READ

The partial order transactions of POMVCC enable highly scalable concurrency control. However, the execution order of transactions is limited by an application or user. For example, consider that the succeeding transaction refers to the result of the preceding transaction. In this case, the HISTORICAL READ, in which the succeeding transaction cannot refer to the result of the preceding transaction, occurs. It is necessary for POMVCC to provide the result of the preceding transaction to the succeeding transaction when the application requires the result of the preceding transaction.

Table VI and Figure 6 provide the definition of HISTORICAL READ. The Tx.2 cannot refer to record X', which Tx.1 updates after the commit of Tx.1. This is a new anomaly. If Tx.1 and Tx.2 are independent transactions, such an anomaly does not occur. However, when the application assumes that the execution order is Tx.1 → Tx.2, an unexpected response occurs. This anomaly of HISTORICAL READ does not occur on MVCC.

TABLE VI. DEFINITION OF HISTORICAL READ

Anomaly	Formula
Historical Read (HR)	$W1[X \rightarrow X'] \ C1 \ B2 \ R2[X]$

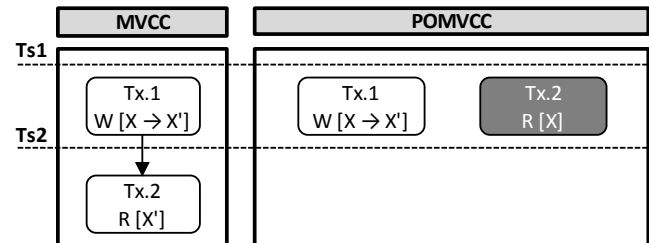


Figure 6. Anomaly of HISTORICAL READ

D. How to avoid HISTORICAL READ

HISTORICAL READ is avoidable if the BTs of the succeeding transaction is larger than the CTs of the preceding transaction. That is, when the same user (DB connection) or the same application executes transactions, the value that is larger than the CTs of the preceding transaction is assigned to the BTs of the succeeding transaction. Therefore, HISTORICAL READ can be avoided.

The avoidance method for the same user (connection-based method) may include false positives. Figure 7 shows the solution of HISTORICAL READ for the connection-based approach. In the worst case, timestamps are updated at every commit. For example, timestamp updates are unnecessary in the independent transactions. However, in the connection-based method, timestamps are always updated during the begin phase of the transactions. As a result, performance degradation is a concern due to there being many false-positive cases.

With the avoidance method for the same application (request-based method), minimum increments of the timestamp, which would preferably be referred to, are set when the application issues transactions. This method can

avoid HISTORICAL READ efficiently because false positives are excluded. However, the interface of a DBMS, such as begin and commit, must be modified, which is a disadvantage of this method. Figure 8 shows the solution of the connection-based method. POMVCC returns a CTs at the commit of Tx.1, and a BTs (= CTs) is set at the begin of Tx.2. As a result, $Tx.1.CTs < Tx.2.BTs$ is established, and Tx.2 can refer to the execution result of Tx.1. We implemented the request-based method shown in Figure 8.

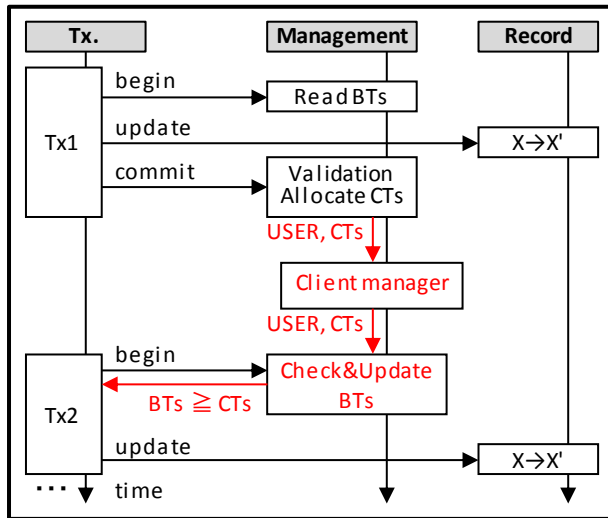


Figure 7. Solution of HISTORICAL READ on connection-based method

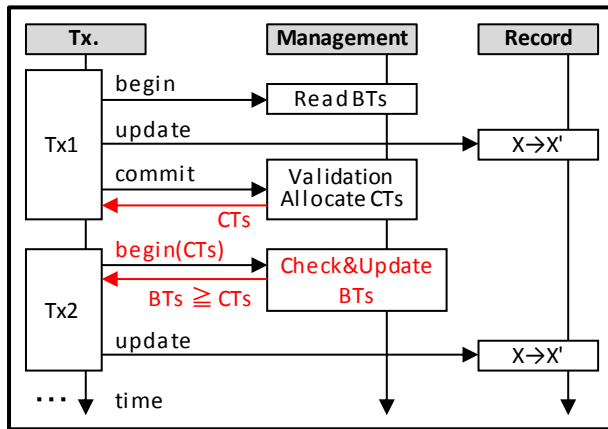


Figure 8. Solution of HISTORICAL READ on request-based method

V. IMPLEMENTATION OF MPDB

We developed an in-memory DBMS called “MPDB” to compare the performance of MVCC and POMVCC. We implemented MVCC and POMVCC on MPDB and evaluated their performance. MPDB is an MVCC-based, lock-free, in-memory DBMS characterized by parallel logs and mixed PCC/OCC [1][2][3]. In this section, we introduce the implementation of MVCC and POMVCC on MPDB.

A. Technical issues

From evaluating the breakdown of TPC-C to organize the DBMS issues on OLTP, buffering (30%), locking (29%) and logging (21%) accounted for 80% of the whole process [11] [26]. The buffering manages temporary data on a DBMS to achieve high-performance by reducing the number of storage accesses. Locking is mainly used for updating when maintaining DBMS consistency by transaction processing. Logging writes log sets to storage to make the transaction results persistent. We aimed to solve these problems on MPDB.

The number of CPU cores and memory size have been increasing. Although the number of cores per CPU has increased rapidly, the CPU frequency is converging to about 3 GHz [35]. Therefore, we must develop high-parallelism for improving performance of tps in line with the technical trend. The memory capacity is also increasing with the momentum exceeding DB size. The data set of OLTP is often several TB or less, and in-memory processing that does not acquire data from storage has become possible. Therefore, we developed an in-memory DBMS called “MPDB” for sustainable and high-performance DBMSs.

B. Design overview

MPDB implements MVCC-based architecture using lock-free on an in-memory DBMS for high-performance and high-scalability OLTP. We implemented lock-free control to avoid degradation of scalability on lock control due to the increased number of CPU cores.

Figure 9 shows a design overview of MPDB. Transaction processing is organized into three phases on MPDB. The first phase is the begin processing and the transaction processing of read and write. The DBMS allocates a timestamp for reference to the transaction during the begin phase and the transaction reads/writes the records using a BTs. The second phase is the validation phase during the commit phase. The processing details are given in Sections V.D and VI. The third phase is the durability phase during the commit phase. The DBMS writes log sets to storage to make the transaction results persistent.

During in-memory processing, client communication and log processing increase in proportion to performance, and interrupt handling becomes a bottleneck. However, load balancing is easy for client communication. The load of interrupt handling can be generally distributed by Receive Side Scaling (RSS) or “irqbalance”. The number of interrupts can be reduced by changing the interrupt handling to polling processing. Log processing must manage the log file sequentially to guarantee the ACID. However, it is not necessary to manage log files physically in one dimension along the time series. A one-dimensional log file is sufficient to produce a logical log file at recovery. Therefore, MPDB implements a mechanism that allows log processing to be executed in parallel by the assigned TxID and timestamp to the transaction log. We implemented asynchronous input/output (I/O) using “libaio” for efficient log processing [36].

The group commit may be a cause of hindering the scalability of log management. We do not implement group commit since random write performance does not become a bottleneck due to the appearance of a high-performance storage such as a solid-state drive or storage class memory.

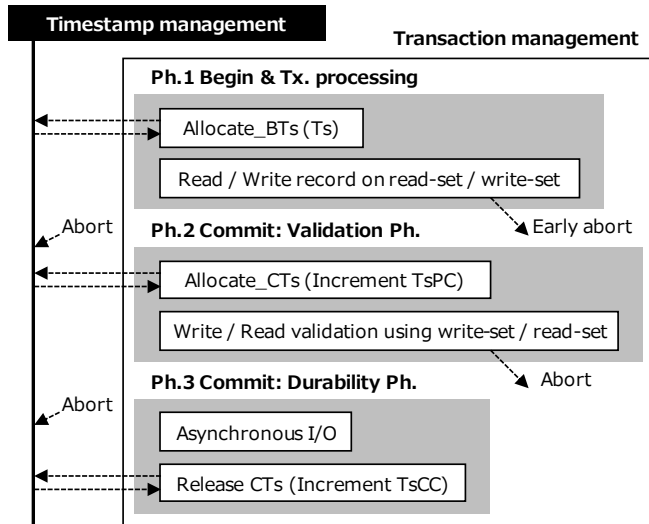


Figure 9. Design overview of MPDB. Ts = Timestamp

C. Data structure

We have to construct the data structure considering the non-uniformity of memory latency by Non-Uniform Memory Access (NUMA) [37]. We divide the data allocation into thread areas, i.e., local areas and a global area in consideration of NUMA on MPDB. Figure 10 shows the overview of data allocation on MPDB. As a premise, MPDB allocates threads of transaction processing to CPU cores.

The thread area manages a work area and log area for transaction processing. Each thread has its own thread area to execute transactions and references/updates another thread area when it executes the validation process, but this is infrequent. Therefore, the thread area should be built in the local area.

We assigned a local area for each CPU. A local area has log-management information to perform log processing for each CPU and is used to expand the thread area.

MPDB assigns common data, such as tables, indexes, and system information, with no locality in the global area. It creates the global area by the NUMA option of “—interleave” to allocate this area and multiple memory to load balance the memory access.

Figure 11 shows the detailed structure of the tables and B-tree index on MPDB. We adopted the linked list for all data structures to implement a lock-free DBMS. MPDB inserts records to update/delete/insert the records for MVCC. We define the rows of the table as a record and the record of update history as a row.

The B-tree index includes nodes and edges. The nodes are arranged in descending order, and edges are arranged in ascending order. MPDB enables bidirectional search by using this index structure. This structure is lock-free since it is made of the linked list.

MPDB also allows the possibility that the index does not refer to the latest record to enable early commit. As shown in Figure 11, transaction processing does not positively change the record pointer of the leaf edge to the pointer of new record when delete Row.1', so that index.col.2 does not necessarily indicate the latest Row.1'. We define this processing method as LATE UPDATE. Therefore, the thread can shorten the serialization point and improve scalability during the commit phase. However, the thread changes the pointer of the record to the latest pointer when referring with the index on LATE UPDATE. The thread can reduce the number of chains of the linked list and achieve fast record access.

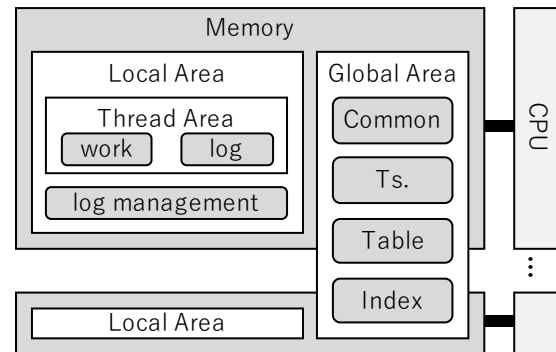


Figure 10. Overview of data structure on MPDB

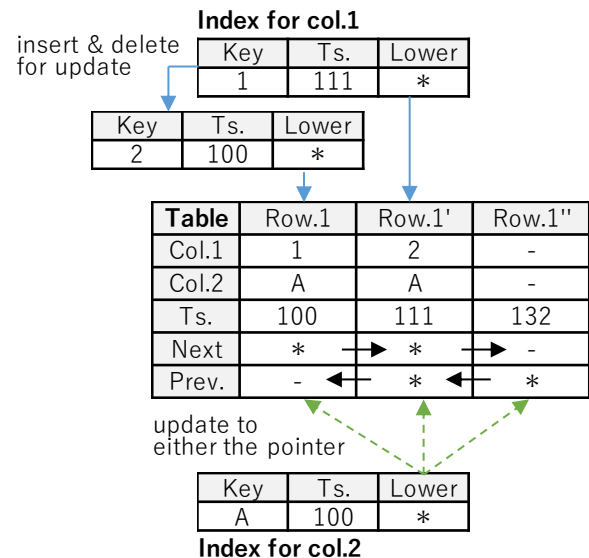


Figure 11. Detailed structure of tables and indexes

D. Transaction management

We now explain the procedure of transaction processing. The state of the transaction is illustrated in Figure 12. MPDB manages the four typical states of transactions. The transaction states can be classified into ACTIVE during the begin phase, PRE-COMMIT at the validation phase during the commit phase, COMMIT at the durability phase during the commit phase and ABORT during the abort phase, as shown in Figures 9 and 12.

Table VII shows the transaction state for each transaction method on MPDB. MPDB implemented mixed OCC/PCC to provide six transaction methods. Generally, long transactions are easily aborted by short transactions; therefore, long transactions can reduce the frequency of the abort when short transactions set OCC and long transactions set PCC.

OCC and PCC are illustrated in Figures 13 and 14, respectively. On OCC, the initial state of the transaction is ACTIVE and the database performs begin processing in the first phase and commit processing in the second–fifth phases. However, on PCC, the initial state of the transaction is PRE-COMMIT and the database performs begin processing in the first phase and the commit processing in the second and third phases. The processing equivalent to write lock is executed with the third phase on OCC and first phase on PCC. That is, since threads can execute write lock during transaction processing on PCC, it is possible to perform record update reservation earlier than OCC. Because of this, MPDB enables the coexistence of long and short transactions.

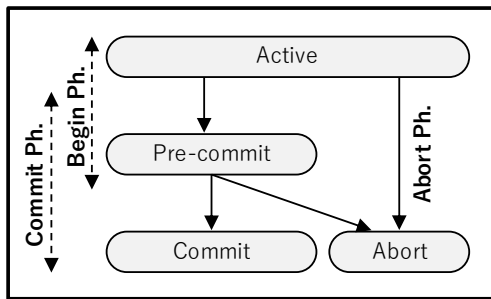


Figure 12. States of transaction

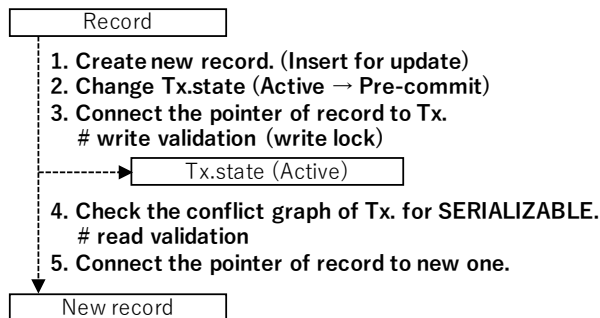


Figure 13. Tx. processing on OCC

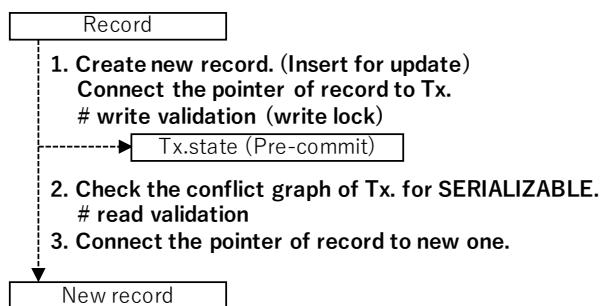


Figure 14. Tx. processing on PCC

TABLE VII. MEHTODS OF TRANSACTION PROCESSING

ISOLATION	BTs. Allocation	Tx.state
PCC.RC	each SQL	state(pre-commit) at begin
PCC.SI	each Tx.	state(pre-commit) at begin
PCC.SERIALIZABLE	each Tx.	state(pre-commit) at begin
OCC.RC	each SQL	state(active) at begin, state(pre-commit) at validation
OCC.SI	each Tx.	state(active) at begin, state(pre-commit) at validation
OCC. SERIALIZABLE	each Tx.	state(active) at begin, state(pre-commit) at validation

VI. IMPLEMENTATION OF POMVCC

The lock used in parallel processing may degrade scalability [6]. In this section, we introduce a lock-free implementation for scalable POMVCC to reduce this degradation.

A. Implementation

We implemented POMVCC to solve the problem of critical section. Previously, the critical section is that the transaction increments a CTs, adapts the CTs to the newest versions and unlocks it during the commit phase. Therefore, Tx.2 waits until the end of Tx.1 to allocate the CTs. Therefore, we divide a timestamp into a BTs and CTs to solve this problem. This is similar to speculative execution. A BTs is the timestamp used for referring to a record. This technique is very common. Table VIII and Figure 15 show the timestamp management and data structure on POMVCC.

We solve the problem of lock for scalability. Generally, transactions increment a CTs during the commit phase in parallel. Therefore, the lock is necessary to obtain the sequential and unique CTs on MVCC. However, POMVCC does not require a unique CTs. That is, a transaction does not increment a CTs during the commit phase on POMVCC. The transaction manager reads a CTs, and the timestamp manager updates it, as shown in Figure 15. On POMVCC, timestamp control is divided into a read process by the transaction manager and a write process by the timestamp manager for lock-free.

Finally, the commit phase is divided into pre-commit and commit. The DBMS must manage committed transactions at the same timestamp on partial order for consistency. Therefore, MPDB implements double counters to manage the state of many transactions at each timestamp. The double counters are Pre-commit Counter at each Timestamp (Ts.PC) and Commit Counter at each Timestamp (Ts.CC). The DBMS can determine the transaction state from the difference between the Ts.PC and Ts.CC. We show the commit process as follows. The Tx.1 reads a CTs,

increments the Ts.PC of the CTs, and adapts the CTs to the newest versions of record during the pre-commit phase. It then increments the Ts.CC of the CTs when the log is completed during the commit phase. Then, Tx.2 does not wait for Tx.1 to execute the commit process. Therefore, POMVCC is highly scalable. Strictly, the atomic processing has critical section for incrementing the Ts.PC or Ts.CC; however, it is very short. The timestamp manager can increment a BTs or CTs anytime when it has detected an anomaly or requirement. For example, if the Ts.PC and Ts.CC are the same, the timestamp manager updates a RTs. That is, the record can be referred to by using this timestamp while maintaining consistency. Table VIII lists the timestamp-management rules on POMVCC.

TABLE VIII. TIMESTAMP-MANAGEMENT RULES

D5. CTs management	
$CTs[a] \rightarrow CTs[a+1] \Leftarrow i$ or any time	
i	$DB(Tx.i \rightarrow Tx.j) \neq DB(Tx.j \rightarrow Tx.i)$
D6. BTs management	
$BTs[b] \rightarrow BTs[b+1] \Leftarrow I$ and II	
I	$BTs[b+1] < CTs[b+1]$
II	$Ts.PC[b+1] = Ts.CC[b+1]$

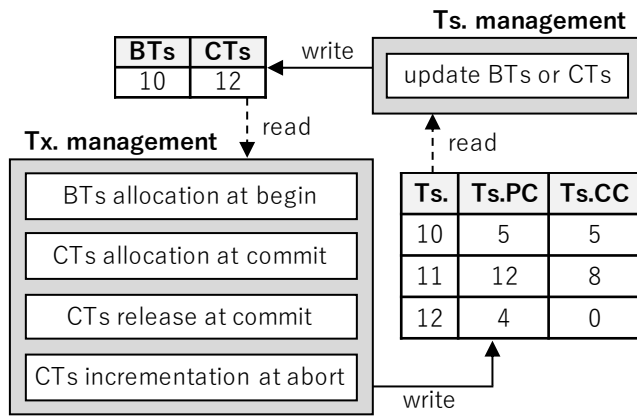


Figure 15. Timestamp management and data structure on POMVCC

B. Log management

Table IX lists the general log-management rules. We define the I/O completion as Completion (Comp). Log management must complete the transaction processing of all CTs (10) if it can complete transaction processing of CTs (11), as shown in Table IX. This rule corresponds to the general cascading protocol for recovery processing.

As a result of separating timestamps into BTs and CTs, this rule became unnecessary on MPDB because the BTs manager guarantees that all readable records persisted, as shown in Table VIII. Log management does not need to control the log execution order, so it can maintain high-scalability.

TABLE IX. LOG-MANAGEMENT RULES

D7. Log management
$Comp(Ts[a]) \rightarrow Comp(Ts[a+1]) \Leftarrow$
$\forall a (Logged(Ts[a]) \leq Logged(Ts[a+1]))$

C. Interface of request-based approach

Figures 16, 17, 18 and 19 illustrate POMVCC. Figure 16 and 17 show the user interface with which a user requests begin, commit or abort to the DBMS, and Figure 18 and 19 show the timestamp interface with which the transaction thread requests any timestamp allocation.

The thread performs the initialization of the data structure and numbering of a BTs during the begin phase. At this time, if the user instructs a transaction to use a timestamp, the timestamp manager increments a CTs up to $Ts + 1$ and increments the BTs up to the timestamps at Allocate_BTs. The timestamp manager stores the transaction-history log when it increments a CTs.

The thread changes the transaction state from ACTIVE to PRE-COMMIT and allocates a CTs from the timestamp manager. When allocating the CTs, the thread increments the Ts.PC to determine the number of transaction processes in the CTs. After that, the thread changes the transaction state from PRE-COMMIT to COMMIT through the validation phase. If the transaction state is COMMIT, the thread stores the log and increments the Ts.CC. If the transaction state is ABORT, the thread decrements the Ts.PC through the abort phase. After completion of the commit phase, the thread provides the result and the CTs to the user.

The thread performs the initialization of the data structure for aborting and incrementing the CTs during the abort phase and provides the result and CTs to the user because the thread increments the CTs to avoid the abort due to refer/update conflict. A transaction must increment a BTs after incrementing a CTs to avoid conflict. Therefore, the user gives the CTs during the begin phase during the retry process. Thus, the transaction can at least avoid the same conflict problem as the previous one.

Finally, the timestamp manager updates the BTs and CTs periodically and asynchronously with transaction processing. This solves the problem in which a user cannot reference update records even after a long time.

// DBMS aborts the Tx.

```

AbortTx () {
    ... abort phase ...
    if (/*DBMS identifies the cause of Ts. on abort.*/)
        CTs = Update_CTs ();
    return ( CTs );
}

```

Figure 16. POMVCC interface 1


```

// DBMS begins the Tx.
BeginTx ( Ts ) {
    ... begin phase ...
    BTs = Allocate_BTs ( Ts );
    return ();
}

// DBMS commits the Tx.
CommitTx () {
    Change Tx.state ( Pre-commit );
    CTs = Allocate_CTs ();
    ... write validation phase ...
    ... read validation phase ...
    Change Tx.state ( Commit / Abort );
    if ( Tx.state = Commit ) { // DBMS can commit the Tx.
        ... durable phase ...
        Increment_TsCC ( CTs );
    } else if ( Tx.state = Abort ) { // DBMS detects the Anomaly.
        CTs = AbortTx ();
        Decrement_TsPC ( CTs );
    }
    return ( CTs ); // Ts. for historical read
}

```

Figure 17. POMVCC interface 2

```

// Tx. is allocated the BTs. at begin for read.
Allocate_BTs ( Ts ) {
    CTs = Read_CTs ();
    while ( CTs ≤ Ts ) {
        CTs = Update_CTs ();
    }
    do {
        BTs = Update_BTs ();
    } while ( BTs < Ts );
    return ( BTs );
}

// Tx. is allocated the CTs. at commit
Allocate_CTs () {
    atomic {
        CTs = Read_CTs ();
        Increment_TsPC ( CTs );
    }
    return ( CTs );
}

```

```

// This function updates the CTs.
Update_CTs () {
    CTs = Increment_CTs ();
    Log_CTs ( CTs-1, Read_TsPC ( CTs-1 ) );
    return ( CTs );
}

```

Figure 18. Ts-management interface 1

```

// This function checks & updates the BTs.
Update_BTs () {
    BTs = Read_BTs ();
    CTs = Read_CTs ();
    // It reads the Ts.Pre-commit Counter (Ts.PC).
    PC = Read_TsPC ( BTs + 1 );
    // It reads the Ts.Commit Counter (Ts.CC).
    CC = Read_TsCC ( BTs + 1 );
    if ( BTs < CTs - 1 && PC = CC )
        BTs = Increment_BTs ();
    return ( BTs );
}

```

Figure 19. Ts-management interface 2

VII. EVALUATION OF PROTOTYPE IMPLEMENTATION

In this section, we compare the performance of MVCC and POMVCC. We implemented MVCC and POMVCC on MPDB and evaluated their performance. In this experiment, we used the industry standard benchmark TPC-C and repeatedly executed the stored procedure calls that model New Order [34].

A. Experimental Environment

Figure 20 depicts the system configuration. Four blade servers were used, i.e., symmetric multiprocessors, and had 8 CPUs (80 cores), 1 TB of memory, and 8 ports of an 8-Gb Fiber Channel (FC). The servers and storage were connected via an FC switch and communicated with FC communication.

In the OS (CentOS 6.5) settings, FC ports were assigned to each CPU to distribute the interrupt overhead of FC communication. Hyper-threading was disabled.

For the MPDB settings, one thread was assigned to one core. This means that MPDB used a maximum of 80 threads. One log file was assigned to one CPU to load balance the logs. The isolation level was SNAPSHOT ISOLATION.

The DB was created on the basis of TPC-C. The number of warehouses was 16 and the size of the DB was 0.72 GB. The item, stock, and order_line tables were used in TPC-C. Indexes were also created for the i_id of the item table, s_w_id and s_i_id of the stock table, and ol_o_id and ol_w_id of the order_line table.

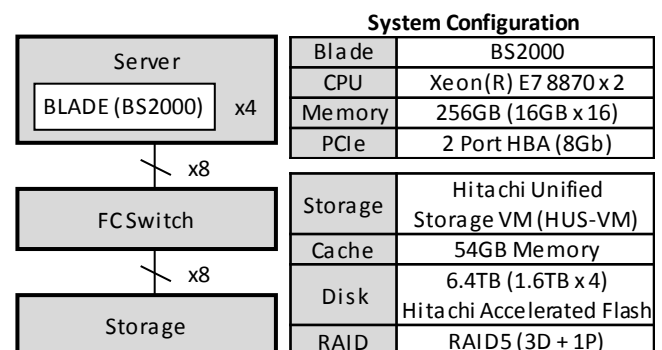


Figure 20. System Configuration

B. Workload

The workload shown in Figure 21 was created on the basis of TPC-C's New Order. The workload simulates the repeatedly executing part of New Order. The processing in Figure 21 was repeated ten times per transaction on average.

1	SELECT	i_price,i_name,i_data
	INTO	:i_price,:i_name,:i_data
	FROM	item
	WHERE	i_id=:ol_i_id
2	SELECT	s_quantity,s_data,s_dist_...
	INTO	:s_quantity,:s_data,:s_dist_...
	FROM	stock
	WHERE	s_i_id=:ol_i_id AND s_w_id=:ol_supply_w_id
3	UPDATE	stock
	SET	s_quantity=:s_quantity
	WHERE	s_i_id=:ol_i_id AND s_w_id=:ol_supply_w_id
4	INSERT	
	INTO	order_line(,,,,)
	VALUES	(,,,,)

While (Repeats 5 ~ 15 times, Ave. 10)

Figure 21. Experiment Workload

C. Experimental Results and Consideration for MPDB

We evaluated MPDB before evaluating POMVCC. We did not use POMVCC to evaluate the basic performance of MPDB. MPDB has various mechanisms but the one most contributing to performance improvement is log parallelization. Therefore, we verified the effects of performance and scalability using parallel log processing. We compared single log processing and parallel log processing and measured the performance and scalability of DBMS with increasing CPU for each log processing.

We compared the performance of single log processing and parallel log processing corresponding to the number of threads. In Figure 22, the x-axis represents the number of threads, and the y-axis represents transactional performance (tps). The performance of parallel log processing increased as the number of threads increased. However, the performance of single log processing decreased as the number of threads increased more than 40 threads. We confirmed that parallel log processing can perform 5.02 times better than single log processing. We also found that I/O interrupt is focused on a specific CPU core by analyzing single log processing with “mpstat” of Linux, as shown in Figure 23. In this figure, the x-axis represents the id of CPU core (0-79), and the y-axis represents time. We confirmed that parallel log processing distributes the load of I/O interrupt.

We also compared the scalability of single log processing and parallel log processing corresponding to the number of threads. In Figure 24, the x-axis represents the number of threads, and the y-axis represents the performance rate on Figure 22 when the performance at 10 threads was assumed as that at 100. In single log processing, the scalability

suddenly deteriorated at 40 threads. However, parallel log processing maintained scalability degradation at less than 15% even with 80 threads.

We confirmed that if the number of CPUs exceeds 2, it is necessary to parallelize log processing.

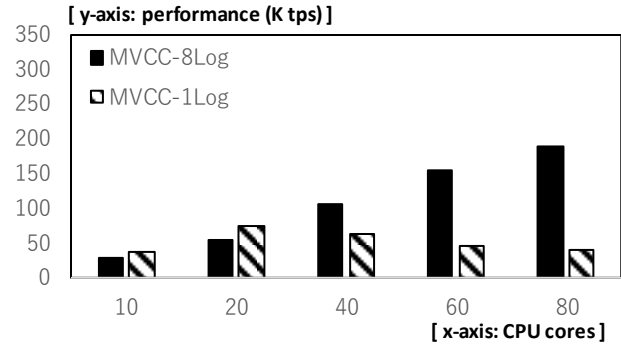


Figure 22. Performance evaluation for log processing

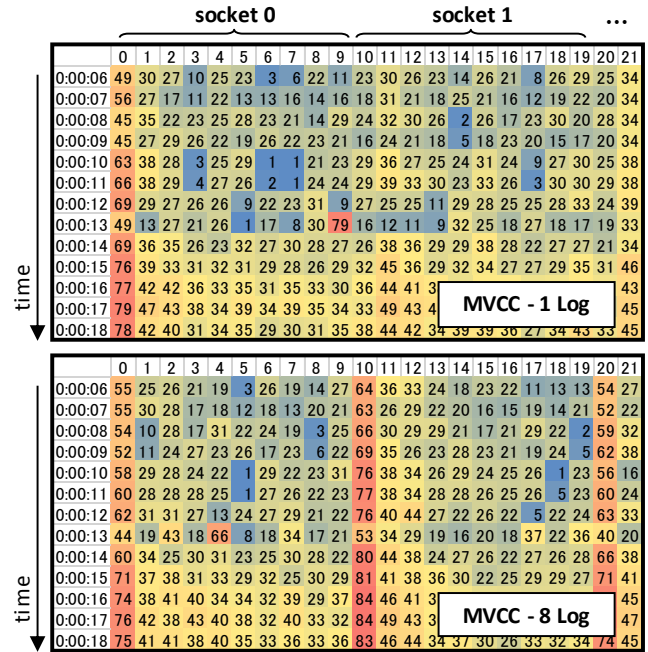


Figure 23. Load of I/O interrupt each CPU core

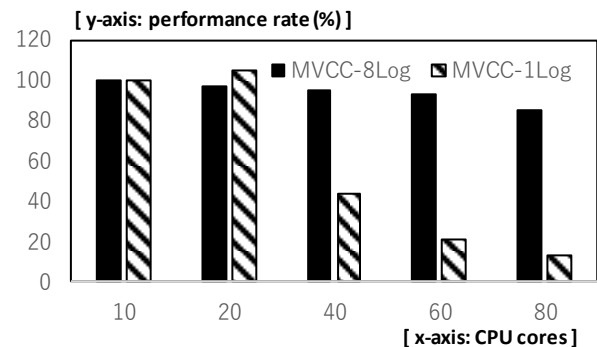


Figure 24. Scalability evaluation for log processing

D. Experimental Results and Consideration for POMVCC

We compared the performance of MVCC (8 logs) and POMVCC (8 logs) corresponding to the number of threads. In Figure 25, the x-axis represents the number of threads, and the y-axis represents tps on increasing conflict rate. The performance of both MVCC and POMVCC increased as the number of threads increased. POMVCC ran 1.36–1.60 times faster than MVCC.

To investigate scalability more precisely, we conducted an experiment in which the number of warehouses changed corresponding to the number of threads. That is, the number of warehouses was ten (DB size was 0.45 GB) when the number of threads was ten and the one was 80 (DB size was 3.61 GB) when the one was 80. The respective experimental results show Figures 26 and 27.

In Figure 26, the x-axis represents the number of threads, and the y-axis represents tps on a fixed conflict rate. The performance of POMVCC on a fixed conflict rate (Figure 26) is higher than the performance on increasing conflict rate (Figure 25). POMVCC on a fixed conflict rate was 1.34 times faster than one on increasing rate. However, MVCC exhibited almost the same performance regarding increasing conflict rate (Figure 25) and regarding the fixed conflict rate (Figure 26). Therefore, POMVCC was 1.63–1.74 times faster than MVCC.

We then compared the scalability of MVCC and POMVCC corresponding to the number of threads at a fixed conflict rate. In Figure 27, the x-axis represents the number of threads, and the y-axis represents the performance rate on Figure 26 when the performance at 10 threads was assumed as that at 100. The scalability of both MVCC and POMVCC slowly decreased as the number of threads increased. The scalability coefficient of MVCC was 87.98–97.96% and that of POMVCC was 94.02–98.32%. POMVCC improved by 6.87% compared with MVCC. This experiment suggests that the scalability coefficient of POMVCC is greater than that of MVCC.

From these experiments, the scalability coefficients of POMVCC and MVCC depended on the size of the DB and number of threads. When the size of the DB was large and the conflict rate of the transaction was low, the scalability coefficient of POMVCC was high, and in all experiments, POMVCC ran faster than MVCC.

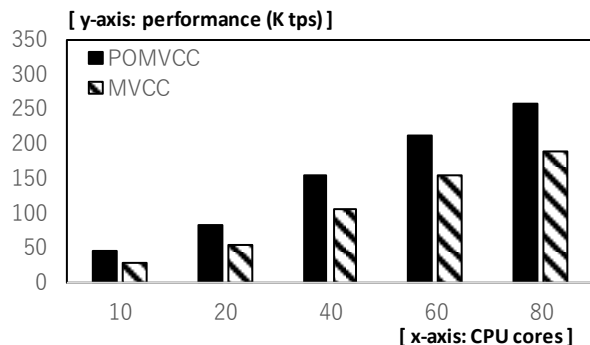


Figure 25. Performance evaluation regarding increasing conflict rate

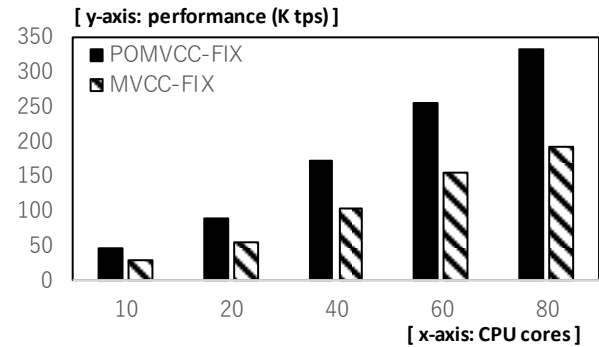


Figure 26. Performance evaluation regarding fixed conflict rate

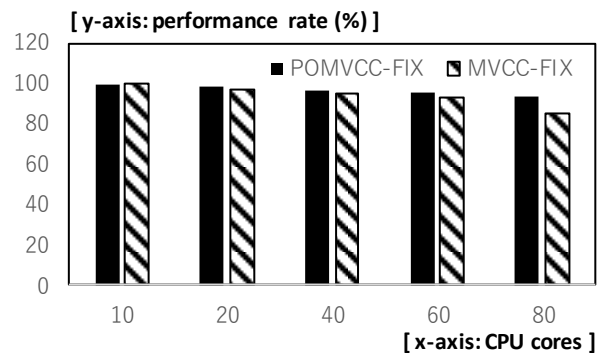


Figure 27. Scalability evaluation regarding fixed conflict rate

VIII. CONCLUSION AND FUTURE WORK

We proposed POMVCC, which maintains the protocol of MVCC and improves performance and scalability of DBMS. POMVCC is focused on the partial order of transactions. The conventional technique provides a timestamp to each transaction, but POMVCC provides a timestamp to multiple transactions. POMVCC reduces the number of timestamps that are updated and improves performance and scalability of DBMS. We discussed the difference in isolation levels between MVCC and POMVCC, as shown in Figure 4.

We implemented and evaluated POMVCC on an in-memory DBMS we developed called “MPDB”, which is an MVCC-based, lock-free, in-memory DBMS that is characterized by parallel logs and mixed PCC/OCC.

We first compared the performance and scalability of MPDB corresponding to the number of threads. The results indicate that the most contributing mechanism to performance improvement was log parallelization. Parallel log processing maintains scalability degradation of less than 15% even with 80 threads. We confirmed that if the number of CPUs exceeds 2, it is necessary to parallelize the log processing.

We then compared the performance and scalability of MVCC and POMVCC corresponding to the number of threads regarding an increasing conflict rate. The performance of POMVCC was 1.36–1.60 times better than that of MVCC. We also compared the performance of MVCC and POMVCC regarding a fixed conflict rate. POMVCC was 1.63–1.74 times faster than MVCC. The

scalability coefficient of MVCC was 87.98–97.96% and that of POMVCC was 94.02–98.32%. The performance of POMVCC improved by 6.87 % compared with MVCC.

The scalability coefficients of POMVCC and MVCC depended on the size of the DB and number of threads. When the size of the DB was large and the conflict rate of the transaction was low, the scalability coefficient of POMVCC was high, and in all experiments, POMVCC ran faster than MVCC.

We implemented POMVCC on MPDB and evaluated it by using SNAPSHOT ISOLATION, for which POMVCC performed better than MVCC. However, the performance trend was unclear because the probability of WRITE SKEW increased on SERIALIZABLE. This occurs when reference and update transactions are executed at the same timestamp. POMVCC increases the number of transactions at the same timestamp. As a result, the number of WRITE SKEWS increases. It is also possible that RW-CONFLICT GRAPH will increase and a large cyclic graph will be created. Therefore, our future work is to implement and evaluate POMVCC by using SERIALIZABLE.

REFERENCES

- [1] Y. Isoda, A. Tomoda, T. Tanaka, and K. Mogi, "Partial Order Multi Version Concurrency Control," DBKDA 2018, The Tenth International Conference on Advances in Databases, Knowledge, and Data Applications, May 2018.
- [2] Y. Isoda, A. Tomoda, K. Ushijima, T. Tanaka, T. Uemura, T. Hanai, and et al., "In-Memory Database Engine for Scale-up System," Forum on Information Technology '15, D-035, 2015 (in Japanese).
- [3] Y. Isoda, K. Ushijima, T. Tanaka, T. Hanai, and K. Mogi, "Proposal of Multi Version Concurrency Control for Partial Order Transaction," Forum on Information Technology '16, D-015, 2016 (in Japanese).
- [4] C. Diaconu, C. Freedman, E. Ismert, P. Larson, P. Mittal, R. Stonecipher, N. Verma, and M. Zwillig, "Hekaton: SQL server's memory-optimized OLTP engine," SIGMOD '13 Proceedings, pp. 1243-1254, 2013.
- [5] H. Kimura, "FOEDUS: OLTP Engine for a Thousand Cores and NVRAM," SIGMOD '15 Proceedings, pp. 691-706, 2015.
- [6] P. Larson, S. Blanas, C. Diaconu, C. Freedman, J. M. Patel, and M. Zwillig, "High-performance concurrency control mechanisms for main-memory databases," Proceedings of the VLDB Endowment, Volume 5 Issue 4, pp. 298-309, 2011.
- [7] T. Wang, and R. Johnson, "Scalable logging through emerging non-volatile memory," Proceedings of the VLDB Endowment, Volume 7 Issue 10, pp. 865-876, 2014.
- [8] V. Sikka, F. Färber, W. Lehner, S. K. Cha, T. Peh, and C. Bornhövd, "Efficient transaction processing in SAP HANA database: the end of a column store myth," SIGMOD '12 Proceedings, pp. 731-742, 2012.
- [9] S. Tu, W. Zheng, E. Kohler, B. Liskov, and S. Madden, "Speedy Transactions in Multicore In-Memory Databases," SOSOP '13 Proceedings, pp. 18-32, Farmington, Pennsylvania, USA, 2013.
- [10] J. Gray, and A. Reuter, "Transaction Processing: Concepts and Techniques," Elsevier, 1992.
- [11] S. Harizopoulos, D. J. Abadi, S. Madden, and M. Stonebraker, "OLTP through the looking glass, and what we found there," SIGMOD '08 Proceedings, pp. 981-992, 2008.
- [12] M. J. Cahill, U. Röhm, and A. D. Fekete, "Serializable isolation for snapshot databases," ACM Transactions on Database Systems, Volume 34 Issue 4, Article No.20, 2009.
- [13] A. Fekete, D. Liarakis, P. O'Neil, and D. Shasha, "Making snapshot isolation serializable," ACM Transactions on Database Systems, Volume 30 Issue 2, pp. 492-528, 2005.
- [14] D. L. Mills, "Internet time synchronization: the network time protocol," IEEE Transactions on Communications, Volume 39, Issue 10, October 1991.
- [15] P. Larson, S. Blanas, C. Diaconu, C. Freedman, J. M. Patel, and M. Zwillig, "High-performance concurrency control mechanisms for main-memory databases," Proceedings of the VLDB Endowment Volume 5 Issue 4, pp. 298-309, 2011.
- [16] J. C. Corbett, J. Dean, M. Epstein, A. Fikes, C. Frost, J. Furman, and et al., "Spanner: Google's Globally Distributed Database," ACM Transactions on Computer Systems, Volume 31 Issue 3, Article No.8, 2013.
- [17] Hewlett Packard, "Memory-Driven Computing," <https://news.hpe.com/content-hub/memory-driven-computing/>, November 2018.
- [18] H. Lim, D. Han, D. G. Andersen, and M. Kasmirsky, "MICA: A Holistic Approach to Fast In-Memory Key-Value Storage," NSDI '14, pp. 429-444, April 2014.
- [19] A. Pavlo, "What are we doing with our lives?," SIGMOD '17 Keynote, <http://www.cs.cmu.edu/~pavlo/slides/pavlo-keynote-sigmod2017.pdf>, November 2018.
- [20] C. Mohan, D. Haderle, B. Lindsay, H. Pirahesh, and P. Schwarz, "ARIES: a transaction recovery method supporting fine-granularity locking and partial rollbacks using write-ahead logging," ACM Transactions on Database Systems, Volume 17 Issue 1, pp. 94-162, 1992.
- [21] D. A. Menascé, and T. Nakanishi, "Optimistic versus pessimistic concurrency control mechanisms in database management systems," Information Systems Volume 7, Issue 1, pp. 13-27, 1982.
- [22] H. T. Kung, and J. T. Robinson, "On optimistic methods for concurrency control," ACM Transactions on Database Systems, Volume 6 Issue 2, pp. 213-226, 1981.
- [23] K. P. Eswaran, J. N. Gray, R. A. Lorie, and I. L. Traiger, "The notions of consistency and predicate locks in a database system," Communications of the ACM, Volume 19 Issue 11, pp. 624-633, 1976.
- [24] I. Pandis, R. Johnson, N. Hardavellas, and A. Ailamak, "Data-oriented transaction execution," Proceedings of the VLDB Endowment, Volume 3 Issue 1-2, pp. 928-939, 2010.
- [25] I. Pandis, P. Tozun, R. Johnson, and A. Ailamaki, "PLP: page latch-free shared-everything OLTP," Proceedings of the VLDB Endowment, Volume 4 Issue 10, pp. 610-621, 2011.
- [26] R. Johnson, I. Pandis, N. Hardavellas, A. Ailamaki, and B. Falsafi, "Shore-MT: a scalable storage manager for the multicore era," EDBT '09 Proceedings, pp. 24-35, 2009.
- [27] P. A. Bernstein, V. Hadzilacos, and N. Goodman, "Concurrency Control and Recovery in Database System," 1987.
- [28] ORACLE, "Oracle Database 12c Release 2," <https://docs.oracle.com/en/database/oracle/oracle-database/12.2/index.html>, November 2018.
- [29] MySQL, "MySQL 5.7 Reference Manual," <https://dev.mysql.com/doc/refman/5.7/en/>, November 2018.
- [30] PostgreSQL, "PostgreSQL 9.6.10 Documentation," <https://www.postgresql.org/docs/9.6/static/index.html>, November 2018.
- [31] H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil, "A Critique of ANSI SQL Isolation Levels," ACM SIGMOD '95 Proceedings, pp. 1-10, San Jose, CA, 1995.

- [32] M. Stonebraker, S. Madden, D. J. Abadi, S. Harizopoulos, N. Hachem, and P. Helland, "The end of an architectural era: (it's time for a complete rewrite)," VLDB '07 Proceedings, pp. 1150-1160, 2007.
- [33] R. Kallman, H. Kimura, J. Natkins, A. Pavlo, A. Rasin, S. Zdonik, and et al., "H-store: a high-performance, distributed main memory transaction processing system," Proceedings of the VLDB Endowment, Volume 1 Issue 2, pp. 1496-1499, 2008.
- [34] The Transaction Processing Council, "TPC-C Benchmark (Version 5.11.0)," <http://www.tpc.org/tpcc/>, November 2018.
- [35] J. L. Hennessy, and D. A. Patterson, "Computer Architecture: A Quantitative Approach," Morgan Kaufmann Publishers.
- [36] E. P. C. Jones, D. J. Abadi, and S. Madden, "Low Overhead Concurrency Control for Partitioned Main Memory Databases," SIGMOD '10 Proceedings, pp. 603-614, June 2010.
- [37] D. Levinthal, "Tutorial: Intel Core i7 and Intel Xeon 5500 Microarchitecture, Optimization and Performance Analysis," 2010 IEEE International Symposium on Performance Analysis of Systems and Software, White Plains, NY, 2010.



www.iariajournals.org

International Journal On Advances in Intelligent Systems

✎ issn: 1942-2679

International Journal On Advances in Internet Technology

✎ issn: 1942-2652

International Journal On Advances in Life Sciences

✎ issn: 1942-2660

International Journal On Advances in Networks and Services

✎ issn: 1942-2644

International Journal On Advances in Security

✎ issn: 1942-2636

International Journal On Advances in Software

✎ issn: 1942-2628

International Journal On Advances in Systems and Measurements

✎ issn: 1942-261x

International Journal On Advances in Telecommunications

✎ issn: 1942-2601