Audrone Lupeikiene
Olegas Vasilecas
Gintautas Dzemyda (Eds.)

Communications in Computer and Information Science 838

Databases and Information Systems

13th International Baltic Conference, DB&IS 2018
Trakai, Lithuania, July 1–4, 2018
Proceedings

Springer

Communications in Computer and Information Science 838

More information about this series at http://www.springer.com/series/7899

Audrone Lupeikiene · Olegas Vasilecas
Gintautas Dzemyda (Eds.)

Databases and Information Systems

13th International Baltic Conference, DB&IS 2018
Trakai, Lithuania, July 1–4, 2018
Proceedings

Springer

Editors

Audrone Lupeikiene
Institute of Data Science and Digital Technologies
Vilnius University
Vilnius
Lithuania

Gintautas Dzemyda
Institute of Data Science and Digital Technologies
Vilnius University
Vilnius
Lithuania

Olegas Vasilecas
Information Systems Department
Vilnius Gediminas Technical University
Vilnius
Lithuania

ISSN 1865-0929 ISSN 1865-0937 (electronic)
Communications in Computer and Information Science
ISBN 978-3-319-97570-2 ISBN 978-3-319-97571-9 (eBook)
https://doi.org/10.1007/978-3-319-97571-9

Library of Congress Control Number: 2018950390

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

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The volume gathers the papers presented at the 13th International Baltic Conference on Databases and Information Systems (DB&IS), which was held in Trakai, Lithuania, during July 1–4, 2018. The conference was organized by the Vilnius University, Vilnius Gediminas Technical University, Lithuanian Academy of Sciences, and Lithuanian Computer Society. The Baltic DB&IS 2018 conference continued the series of biennial conferences, which have been held in Trakai (1994), Tallinn (1996, 2002, 2008, 2014), Riga (1998, 2004, 2010, 2016), and Vilnius (2000, 2006, 2012). Since its inception in 1994 by Prof. Janis A. Bubenko Jr. (Royal Institute of Technology and Stockholm University, Sweden) and Prof. Arne Sølvberg (Norwegian University of Science and Technology, Norway) the Baltic DB&IS conference has become an international forum for researchers and developers in the field of databases, information systems, and related areas. This year's submissions represented the five continents. The objective of this conference series is to bring together researchers, practitioners, and PhD students and provide an environment in which they can present their work, discuss current issues, and exchange the ideas.

This volume of Springer's *Communications in Computer and Information Science* (CCIS) continues the initiative, which began at Baltic DB&IS 2016 (held in Riga), of publishing the conference proceedings in this series. For Baltic DB&IS 2018 we received 69 papers from 15 countries. Each paper was reviewed by at least three Program Committee members. The international Program Committee consisted of 91 members from 31 countries. The evaluation process resulted in the selection of 24 papers (acceptance rate of 34.7%), which were accepted for presentation at the conference and publication in this conference proceedings.

The original research results presented in the papers concern well-established fields such as database systems; architectures and quality aspects of information systems; requirements engineering; ontology engineering; business process modeling; applications and case studies; as well as the novel fields, such as cyber-physical systems; Internet of Things; big data processing; big data analysis and semantics; cognitive computing techniques using artificial intelligence, sophisticated pattern, and speech recognition. This volume also includes two plenary session and three keynote talks.

The organizers were pleased to invite four leading experts for keynote addresses: Marlon Dumas, University of Tartu, Estonia; Marko Bajec, University of Ljubljana, Slovenia; Milan Zdravković, University of Niš, Serbia; and Inguna Skadiņa, University of Latvia, Latvia. The conference was preceded by a one-day doctoral consortium chaired by Raimundas Matulevičius, University of Tartu, Estonia.

Finally, as editors of this volume, we express our deep gratitude to the members of the Program Committee and the external reviewers for their time, comments, and constructive evaluations. We also sincerely thank the members of the international Steering Committee for their continued support of the conference. We would like to thank everyone from the Organizing Committee, especially Laima Paliulioniene, for their time and dedication to help make this conference a success. Special thanks to Prof. Albertas Caplinkas for his sharing of experience and valuable support. We are grateful to the authors and all the participants who truly made the conference a success.

June 2018

Audrone Lupeikiene
Olegas Vasilecas
Gintautas Dzemyda

Organization

Steering Committee

Janis Bubenko (Honorary Member)	Royal Institute of Technology and Stockholm University, Sweden
Arne Sølvberg (Honorary Member)	Norwegian University of Science and Technology, Norway
Guntis Arnicāns	University of Latvia, Latvia
Juris Borzovs	University of Latvia, Latvia
Albertas Čaplinskas	Vilnius University, Lithuania
Jānis Grundspeņķis	Riga Technical University, Latvia
Hele-Mai Haav	Tallinn University of Technology, Estonia
Ahto Kalja	Tallinn University of Technology, Estonia
Mārīte Kirikova	Riga Technical University, Latvia
Audronė Lupeikienė	Vilnius University, Lithuania
Raimundas Matulevičius	University of Tartu, Estonia
Tarmo Robal	Tallinn University of Technology, Estonia
Olegas Vasilecas	Vilnius Gediminas Technical University, Lithuania

General Chair

Gintautas Dzemyda	Vilnius University, Lithuania

Program Co-chairs

Audronė Lupeikienė	Vilnius University, Lithuania
Olegas Vasilecas	Vilnius Gediminas Technical University, Lithuania

Doctoral Consortium Chair

Raimundas Matulevičius	University of Tartu, Estonia

Program Committee

Rajendra Akerkar	Western Norway Research Institute, Norway
Mehmet Akşit	University of Twente, The Netherlands
Guntis Arnicāns	University of Latvia, Latvia
Liz Bacon	University of Greenwich, UK
Marko Bajec	University of Ljubljana, Slovenia
Romas Baronas	Vilnius University, Lithuania
Josef Basl	University of West Bohemia, Czech Republic
Jānis Bičevskis	University of Latvia, Latvia

Mária Bieliková	Slovak University of Technology in Bratislava, Slovakia
Juris Borzovs	University of Latvia, Latvia
Boštjan Brumen	University of Maribor, Slovenia
Robert Buchmann	Babeş-Bolyai University, Romania
Albertas Čaplinskas	Vilnius University, Lithuania
Grzegorz Chmaj	University of Nevada, USA
Christine Choppy	University of Paris, France
Vytautas Čyras	Vilnius University, Lithuania
Robertas Damaševičius	Kaunas University of Technology, Lithuania
Dalė Dzemydienė	Vilnius University, Lithuania
Johann Eder	University of Klagenfurt, Austria
Flavius Frasincar	Erasmus University Rotterdam, The Netherlands
Wojciech Froelich	University of Silesia, Poland
John Gammack	Zayed University, United Arab Emirates
Jorge Esparteiro Garcia	Polytechnic Institute of Viana do Castelo, Portugal
Jānis Grabis	Riga Technical University, Latvia
Jānis Grundspeņķis	Riga Technical University, Latvia
Saulius Gudas	Vilnius University, Lithuania
Sevinc Gulsecen	Istanbul University, Turkey
Giancarlo Guizzardi	Free University of Bozen-Bolzano, Italy; Federal University of Espírito Santo (UFES), Brazil
Hele-Mai Haav	Tallinn University of Technology, Estonia
Władysław Homenda	Warsaw University of Technology, Poland
Zbigniew Huzar	Wroclaw University of Technology, Poland
Ali Hakan Isik	Mehmet Akif Ersoy University, Turkey
Mirjana Ivanović	University of Novi Sad, Serbia
Hannu Jaakkola	Tampere University of Technology, Finland
Andrzej Jardzioch	West Pomeranian University of Technology, Poland
Ignacy Kaliszewski	Systems Research Institute, Polish Academy of Sciences, Poland
Ahto Kalja	Tallinn University of Technology, Estonia
Dimitris Karagiannis	University of Vienna, Austria
Mārīte Kirikova	Riga Technical University, Latvia
Dmitry Korzun	Petrozavodsk State University, Russia
Manolis Koubarakis	National and Kapodistrian University of Athens, Greece
Olga Kurasova	Vilnius University, Lithuania
Michael Lang	National University of Ireland Galway, Ireland
Dejan Lavbič	University of Ljubljana, Slovenia
Innar Liiv	Tallinn University of Technology, Estonia
Grazia Lo Sciuto	University of Catania, Italy
Natalia Loukachevitch	Lomonosov Moscow State University, Russia
Hui Ma	Victoria University of Wellington, New Zealand
Alexander Mädche	Karlsruhe Institute of Technology, Germany
Ka Lok Man	Xi'an Jiaotong-Liverpool University, China

Rainer Manthey	University of Bonn, Germany
Sanda Martinčić-Ipšić	University of Rijeka, Croatia
Saulius Maskeliūnas	Vilnius University, Lithuania
Raimundas Matulevičius	University of Tartu, Estonia
Martin Nečaský	Charles University, Czech Republic
Laila Niedrite	University of Latvia, Latvia
Christophoros Nikou	University of Ioannina, Greece
Boris Novikov	St-Petersburg University, Russia
Jyrki Nummenmaa	University of Tampere, Finland
Vladimir A. Oleshchuk	University of Agder, Norway
Oscar Pastor	Polytechnic University of Valencia, Spain
Justyna Patalas-Maliszewska	University of Zielona Gora, Poland
Ivan Piletski	Belarusian State University of Informatics and Radioelectronics, Belarus
Jaroslav Pokorný	Charles University in Prague, Czech Republic
Boris Rachev	Technical University of Varna, Bulgaria
Jolita Ralyte	University of Geneva, Switzerland
Šarūnas Raudys	Vilnius University, Lithuania
Tarmo Robal	Tallinn University of Technology, Estonia
Radi Romansky	Technical University of Sofia, Bulgaria
José Raúl Romero	University of Cordoba, Spain
Gunter Saake	Otto von Guericke University of Magdeburg, Germany
Kurt Sandkuhl	University of Rostock, Germany
Simonas Šaltenis	Aalborg University, Denmark
André Schekelmann	Niederrhein University of Applied Science, Germany
Michał Śmiałek	Warsaw University of Technology, Poland
Lovro Šubelj	University of Ljubljana, Slovenia
Janis Stirna	Stockholm University, Sweden
João Manuel R. S. Tavares	University of Porto, Portugal
Kuldar Taveter	Tallinn University of Technology, Estonia
Ernest Teniente	Polytechnic University of Catalonia, Spain
Jaak Tepandi	Tallinn University of Technology, Estonia
Emiliano Tramontana	University of Catania, Italy
Peter Thanisch	University of Tampere, Finland
Michal Valenta	Czech Technical University in Prague, Czech Republic
Juan Manuel Vara	University Rey Juan Carlos, Spain
Damjan Vavpotič	University of Ljubljana, Slovenia
Benkt Wangler	Stockholm University, Sweden
Katarzyna Wasielewska	State University of Applied Sciences in Elblag, Poland
Hans Weigand	Tilburg University, The Netherlands
Jelena Zdravkovic	Stockholm University, Sweden
Pavel Zezula	Masaryk University, Czech Republic

Additional Reviewers

Dominik Bork	University of Vienna, Austria
Durand Gabriel Campero	Otto von Guericke University Magdeburg, Germany
Marco Franceschetti	Alpen-Adria-Universität Klagenfurt, Austria
Angelos P. Giotis	University of Ioannina, Greece
Vimal Kunnummel	University of Vienna, Austria
Kestutis Normantas	Vilnius Gediminas Technical University, Lithuania
Ruben Salado	University of Cordoba, Spain
Titas Savickas	Vilnius Gediminas Technical University, Lithuania
Giorgos Sfikas	University of Ioannina, Greece
Tatjana Welzer	University of Maribor, Slovenia

Organizing Chair

Saulius Maskeliūnas	Vilnius University, Lithuanian Computer Society

Publicity Chair

Saulius Gudas	Vilnius University, Lithuania

Finance Chair

Snieguolė Meškauskienė	Vilnius University, Lithuania

Webmaster

Laima Paliulionienė	Vilnius University, Lithuania

Organizing Committee

Dalė Dzemydienė	Vilnius University, Lithuania
Justinas Janulevičius	Vilnius Gediminas Technical University, Lithuania
Kristina Lapin	Vilnius University, Lithuania
Jolanta Miliauskaitė	Vilnius University, Lithuania
Laima Paliulionienė	Vilnius University, Lithuania
Raimundas Savukynas	Vilnius University, Lithuania
Aidas Žandaris	Lithuanian Computer Society, Lithuania

Supporting Institutions and Partners

Vilnius University

IEEE

Springer

Go Vilnius

Contents

Plenary Session

Invited Talks

Information Systems Engineering

Knowledge and Ontologies

Advanced Database Systems

Big Data Analysis and Processing

Cognitive Computing

Applications and Case Studies

Plenary Session

Data Science and Advanced Digital Technologies

Gintautas Dzemyda(✉)

Institute of Data Science and Digital Technologies, Vilnius University,
Akademijos St. 4, 04812 Vilnius, Lithuania
gintautas.dzemyda@mii.vu.lt

Abstract. The most topical challenges in data science are highlighted. The activities of Vilnius University Institute of Data Science and Digital Technologies are introduced. The institute pretends to solve at least a part of problems arising in this field, first of all, cognitive computing, blockchain technology, development of cyber-social systems and big data analytics gintautas.

Keywords: Data science · Challenges · Digital technologies
Computer science

1 Introduction

Every decade, the new research areas appear. We see such rapid evolution in data analysis and computer science. There are new areas of research and new terms, such as data mining, knowledge discovery, deep learning etc. In general, these are some generalizations of previously known fields. However, when generalizations become significant, not only a new term appears, but it becomes an impulse for fast development of the field. Data science is one of such new terms and seeks for more exact definition and purification. Despite this, researchers with similar knowledge agree with the basic concepts of this field.

2 The Concept of Data Science

There is no unified concept of Data Science (DS). A formal definition of data science is indefinite. We can find many different viewpoints to DS depending on the context. Extensive discussions on this topic including evolutionary and fundamental aspects may be found in a large number of references [1–11]. In all cases, DS is a multi-disciplinary subject with data mining, big data, data analytics, machine learning and discovery of data insights. DS combines three highly iterating research fields: mathematics/statistics/operation research, computer science and digital technologies used to study and perceive data.

The statistics and operation research disciplines are very associated with data science, because data science comprises the science of planning for data, acquisition, management, analysis of data, and inference from data [5]. Most of the principles, frameworks, and methodologies that encompass statistics and operation research were

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 3–7, 2018.
https://doi.org/10.1007/978-3-319-97571-9_1

originally developed as solutions to practical problems. Moreover, data science today goes beyond specific areas like data mining and machine learning or whether it is the next generation of statistics [1]. Computer science consists of different technical concepts such as programming languages, algorithm design, software engineering, computer-human interaction and the process of computation [6]. Computer science principal areas include database systems, networks, security, a theory of informatics and bioinformatics. Advanced digital technologies cover data mining, big data analytics, data visualization, high performance computing, in-memory computation (in-memory key value stores), cloud computing, social computing, neurocomputing, deep learning, machine learning, data feeds, overlay networks, cognitive computing, crowdsourcing, log analysis, container-based virtualization, block chaining, life-time value modelling.

In particular, DS is related to visualizations, statistics, pattern recognition, neurocomputing, machine learning, artificial intelligence, databases and data processing, data mining, knowledge discovery in databases. DS finds new fields of applications: chemical engineering, biotechnology, building energy management, materials microscopy, geographic research, learning analytics, radiology, metal design, ecosystem homeostasis investigation [13–22] and many others.

3 Challenges

Currently, the data science still is very young and immature discipline. There are a lot of complex challenges. In our opinion, the most important challenges in data science currently are as follows:

- seamless integration of technologies supporting data sciences into complex cyber-physical-social systems,
- further development of data-driven intelligence methods and methodologies,
- the mingling of the "physical & digital worlds" within the container of Data Science, AI, and Machine Learning.

4 Institute of Data Science and Digital Technologies of Vilnius University

The history of the Institute of Data Science and Digital Technologies [12] starts in 1956. Its name evolved from the Institute of Physics and Mathematics, to the Institute of Mathematics and Cybernetics, and Institute of Mathematics and Informatics. For long time, the Institute belonged to the Lithuanian Academy of Science. Since 2010, it became a part of Vilnius University. Since October 1, 2017, it has the new name – Institute of Data Science and Digital Technologies.

DMSTI was set up to pursue long-term research for the economy of Lithuania and international cooperation. The main fields of DMSTI activities:

- scientific research and experimental development,
- studies (doctoral studied in informatics, informatics engineering, and mathematics; bachelor studies in information system engineering),

- scientific organizational work (conferences, e.g., “Data Analysis Methods for Software Systems” DAMSS’2018, https://www.mii.lt/damss/),
- publishing (“Informatica”, https://www.mii.lt/informatica/, “Baltic Journal of Modern Computing”, “Nonlinear Analysis. Modelling and Control”, “Informatics in Education”, “Modern Stochastics: Theory and Applications”, “Olympiads in Informatics”),
- education (e.g., Bebras Contest on Informatics and Computer Fluency, https://bebras.org/).

Institute has eight research groups, whose names represent research interests of the teams:

1. Blockchain Technologies,
2. Cognitive Computing,
3. Cyber-Social Systems Engineering,
4. Education Systems,
5. Global Optimization,
6. Image and Signal Analysis,
7. Operations Research,
8. Statistics and Probability.

Blockchain Technologies Group. Research fields: - challenges and opportunities of the blockchain technologies for computer science research; systematization of existing blockchain platforms devoted to the scientific research; exploring blockchain technology applicability for the operation research and other related fields; developing blockchain-based solutions for operation research business applications.

Cognitive Computing Group. Research fields: - artificial neural networks; - big data; - bioinformatics; - data mining; - deep learning; - multi-objective optimization; - image analysis, - medical image processing; - internet data mining; - fractal dimensionality; - speech emotion recognition; - local optimization methods; - machine learning; - medical data analysis and decision support; - multiple criteria decision support; - visualization of multidimensional data.

Cyber-Social Systems Engineering Group. Research fields: - theoretical foundations of information systems; - domain causal dependencies modelling for software engineering; - model based applications development for different types of domains (enterprises, Internet of Things, smart systems, etc.); - knowledge-based development of cyber-physical-social systems; - process mining; - automated deduction; - knowledge analysis methods; - deductive systems.

Education Systems Group. Research fields: - application of intelligent technologies in education; - computer science (Informatics) education research; - computing engineering education research; - personalized learning; - software localization; - technology enhanced learning.

Global Optimization Group. Research fields: - optimization and high-performance computing; - nonlinear control methods; - time delay estimation.

Image and Signal Analysis Group. Research fields: - speech and language processing (speech signal modelling: autoregressive/linear prediction, fractal-based nonlinear modelling, source-filter models; speech recognition: segment-based speech recognition; speech emotion recognition: hierarchical classification of speech emotions, speech emotion features, feature selection; estimation of speech quality: acoustic analysis for quality estimation; estimation of vocal fold functionality; estimation of phonation level; signal model parameter assessment; development of sound mathematical models); - image and video signal processing (object recognition and segmentation; movement detection; optical flow analysis; image restoration; machine and deep learning methods development for image analysis; medical image analysis: ophthalmic, microscopy, cell classification, MRI data processing); - deep learning and big data analysis (abnormal marine traffic detection; tumor heterogeneity evaluation).

Operations Research Group. Research fields: - creation and computational realization of complex simulation models in epidemiology, education, economics, and energy supply systems and of various other origin with uncertainty; - operations research; - statistical simulation; - stochastic programming; - swarm intelligence.

Statistics and Probability Group. Research fields: - statistical inference for long memory processes; - statistical hypothesis testing; - heavy tails; - aggregation; - random fields; - self-similar processes; - Levy processes; - rough paths; - random Hamiltonians; - finite population statistics and statistical analysis of data; - extremal problems in harmonic analysis; - random graphs; - combinatorics; - discrete mathematics; - algebraic geometry.

5 Conclusions

The data science can produce valuable results only in combination with a number of advanced digital technologies. In addition, these technologies should be integrated into a variety of computing systems taking into account architecture and other peculiarities of those systems. Consequently, there is also a broad field for research for computer scientists, software engineering, and other informatics-related scientists. Vilnius University Institute of Data Science and Digital technologies pretends to solve at least a part of problems arising in this field, first of all, cognitive computing, blockchain technology, development of cyber-social systems and big data analytics.

References

1. Cao, L.: Data science: challenges and directions. Commun. ACM **60**(8), 59–68 (2017)
2. Matsudaira, K.: The science of managing data science. Commun. ACM **58**(6), 44–47 (2015)
3. Donoho, D.: 50 Years of Data Science. Computer Science and Artificial Intelligence Laboratory, MIT, Cambridge. http://courses.csail.mit.edu/18.337/2015/docs/50YearsDataScience.pdf. Accessed 1 May 2018
4. American Statistical Association Undergraduate Guidelines Workgroup, Curriculum Guidelines for Undergraduate Programs in Statistical Science. https://www.amstat.org/asa/files/pdfs/EDU-guidelines2014-11-15.pdf. Accessed 1 May 2018
5. What is Data Science? http://dsc.ucsd.edu/node/2. Accessed 1 May 2018
6. Computer Science vs Data Science – find out the best 8 comparisons. https://www.educba.com/computer-science-vs-data-science/. Accessed 1 May 2018
7. Huber, P.J.: Data Analysis: What Can Be Learned from the Past 50 Years. Wiley, New York (2011)
8. Blum, A., Hopcroft, J., Kannan, R.: Foundations of Data Science. https://www.cs.cornell.edu/jeh/book.pdf. Accessed 1 May 2018
9. Berman, F., Rutenbar, R., Hailpern, B., et al.: Realizing the potential of data science. Commun. ACM **61**(4), 67–72 (2018)
10. Cao, L.: Data science: a comprehensive overview. ACM Comput. Sur. **50**(3) (2017). Article number 43
11. Gil Press: A very short history of data science. Forbes. https://www.forbes.com/sites/gilpress/2013/05/28/a-very-short-history-of-data-science/#4d91ab9455cf. Accessed 1 May 2018
12. Institute of Data Science and Digital Technologies. https://www.mii.lt. Accessed 1 May 2018
13. Beck, D., Pfaendtner, J., Carothers, J., et al.: Data science for chemical engineers. Chem. Eng. Prog. **113**(2), 21–26 (2017)
14. Selvarajan, E., Punya Swaroop, S.: A review on data science in biotechnology. Res. J. Pharm. Biol. Chem. Sci. **8**(2), 562–567 (2017)
15. Molina-Solana, M., Ros, M., Dolores Ruiz, M., et al.: Data science for building energy management: a review. Renew. Sustain. Energy Rev. **70**, 598–609 (2017)
16. Voyles, P.M.: Informatics and data science in materials microscopy. Curr. Opin. Solid State Mater. Sci. **21**(3), 141–158 (2017)
17. Andrienko, G., Andrienko, N., Weibel, R.: Geographic data science. IEEE Comput. Graph. Appl. **37**(5), 15–17 (2017)
18. Klasnja-Milicevic, A., Ivanovic, M., Budimac, Z.: Data science in education: big data and learning analytics. Comput. Appl. Eng. Educ. **25**(6), 1066–1078 (2017)
19. Aerts, H.J.W.L.: Data science in radiology: a path forward. Clin. Cancer Res. **24**(3), 532–534 (2018)
20. Verma, A.K., French, R.H., Carter, J.L.W.: Physics-informed network models: a data science approach to metal design. Integr. Mater. Manufact. Innov. **6**(4), 279–287 (2017)
21. Morkunas, M., Treigys, P., Bernataviciene, J., Laurinavicius, A., Korvel, G.: Machine learning based classification of colorectal cancer tumour tissue in whole-slide images. Informatica **29**(1), 75–90 (2018)
22. Kikuchi, J., Ito, K., Date, Y.: Environmental metabolomics with data science for investigating ecosystem homeostasis. Prog. Nucl. Magn. Reson. Spectrosc. **104**, 56–88 (2018)

A Light Insight into Latvian and Lithuanian ICT Terminology: Whether Kindred Language Imply Kindred Terminology?

Juris Borzovs(✉)

Faculty of Computing, University of Latvia,
Raiņa bulvāris 19, Riga LV-1586, Latvia
juris.borzovs@lu.lv

Abstract. Lithuanian and Latvian are two closely related languages, the only two of the Baltic branch of Indo-European languages. They are quite similar and share a great deal of vocabulary and grammar features, but not close enough to make conversation possible. The paper reveals that commonalities between Latvian and Lithuanian information and communication technologies (ICT) terms are mostly due internationalisms, there are only small proportion of terms with common Baltic word-roots; influence of English language in Latvian and Lithuanian ICT terminology is moderate, if not minor; deliberately or unawares, Lithuanian terminologists follow the same rules as their Latvian counterparts.

Keywords: Latvian · Lithuanian
Information and communications technology · Terminology · Similarity

1 Answer to Frequently Asked Question

For non-Baltic readers: Lithuanian and Latvian are two closely related languages, the only two of the Baltic branch of Indo-European languages. They are quite similar and share a great deal of vocabulary and grammar features, but not close enough to make conversation possible. Comprehensive treatment of closeness of the languages is given in [1]. However, we will restrict ourselves with few examples.

In Lithuanian and Latvian languages, there are plenty of the same word-roots,

cf. : akis: acs, anglis: ogle [uogle], aš: es, balandis: balodis [baluodis], baltas: balts, bėda : bēda, broils : brālis, būti: būt, ežeras: ezers, diena: diena, Dievas : Dievs, drąsus: drošs [druoš], duoti: dot [duot], galva: galva, gimti: dzimt, epušė: apse, eiti: iet, ėsti : ēst, ežeras: ezers, ežys: ezis, gabalas: gabals, galva: galva, gegužė “gegutė” : dzeguze, gulbė: gulbis, jūra: jūra, jūs: jūs, kanklės: kokle [kuokle], kitas: cits, koja: kāja, lapas: lapa, lapė: lapsa, ledas: ledus, lietus: lietus, lūpa: lūpa, mes: mēs, motina : māte, obuolys : ābols [ābuols], oda : āda, pavasaris: pavasaris, pelė: pele, pienas: piens, ranka: roka [ruoka], ruduo: rudens, saulė: saule, sliekas: slieka, stirna: stirna, sveikas: sveiks, šuo: suns, tėvas : tēvs, tu: tu, upė: upe, vanduo: ūdens, vasara: vasara, velnias: velns, vyras: vīrs, višta: vista, žemė: zeme, žiema: ziema, žinoti : zināt, žolė: zāle et al.

Happens that word-roots are misleading, cf. elnias : briedis and briedis : alnis.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 8–11, 2018.
https://doi.org/10.1007/978-3-319-97571-9_2

This same abundance of words adds words that in one of the languages (or both) have the status of vernacularism (regional word) or archaism,

cf. blezdinga: bezdelīga "kregždė", cyrulis : cīrulis "vyturys", jeknos : aknas "kepenys", kanduolas : kodols "branduolys", krupis : krupis "rupūžė", medžias : mežs "miškas", notrė : nātre "dilgėlė", pylė : pīle "antis", spėkas : spēks "jėga", vetušas : vecs "senas".

Algirdas Sabaliauskas in [2] has collected more than 400 true Baltic (Lithuanian, Latvian, Prussian) word-roots. Much more he added counting common Lithuanian and Latvian words of Indo-European or Balto-Slavonic origin.

2 The Ten Commandments of Latvian Terminology Development

When localizing ICT terms into Latvian, the following ten guidelines were created and observed in the localization process [3, 4]:

1. One term in the original language should correspond to one specific term in the target language.
2. Differing terms in the original language should be given differing terms in the target language.
3. If a term is ambiguous in the original language, a word with a similar range of ambiguity should be chosen in the target language.
4. When coining a neologism, observe its suitability in the corresponding term system and similarity with related and analogic terms.
5. One should choose a term's equivalent so that, when translating it back to the original language, the same original word is the clear choice.
6. When borrowing a word, pay heed to how well it fits into the target language semantically, phonetically, and morphologically.
7. When faced with a choice between international borrowings and native words, preference is given to native words.
8. Do not change, without a sound basis, a word already used in practice.
9. More attention should be paid to words widely used by the general public. They should be short, precise, euphonious, and easy to understand.
10. None of the aforementioned principles shall be made absolute.

These commandments could be "translated" into flowcharts [3, pp. 88–89] thus making terminology localization work more disciplined.

3 The Most Frequently Used ICT Terms in English, Latvian and Lithuanian

According to methodology described in [4] we selected the most frequently used ICT terms from bilingual English-Latvian corpus. Made presumption that Lithuanian ICT terms usage has similar frequency we added Lithuanian terms to the selection (Table 1).

Table 1. 41 frequently used English ICT terms found in the English-Latvian bilingual corpus and their Latvian and Lithuanian counterparts in EuroTermBank [5]

EN	LV	LT
Mode	režīms	režimas
Warning	brīdinājums	Įspėjìmas, pérspėjimas
Window	logs	langas
Click	klikšķis	spràgtelėti, paspáusti
Key	atslēga; taustiņš	raktas; klavišas
File	datne, fails	failas
Service	pakalpojums, serviss	paslauga, tarnyba, sèrvisas
Download	lejuplādēt	atsisiųsti, įkráuti
Information	informācija	informãcija
Data	dati	dúomenys
System	sistēma	sistemà, įrenginỹs, komplèksas
Help	palīdzība	pagálba, konsultãcija
Search	meklēt	ieškóti
User	lietotājs	vartótojas
Computer	dators	kompiuteris
Internet	internets	internètas
Location	vieta	vietà
Security	drošība	apsaugà
Program	programma	programà
Web	tīmeklis	žiniãtinklis
Message	ziņojums	pranešimas
Link	saite	grandìs; jungė̃
Code	kods	kòdas
Form	veidlapa	blankas
Online	tiešsaiste	prijungtìnis
Software	programmatūra	programinė įranga
Folder	mape	ãplankas
Network	tīkls	tiñklas
Application	lietotne	táikomoji programà
Field	lauks	laukas
Comment	komentārs	komentãras
Guest	viesis	svẽčias
Format	formāts	fòrma; formãtas
Table	tabula	lentẽlė
Bit	bits	bitas
Card	karte	kortà
PC	personālais dators	asmeninis kompiuteris
Display	displejs; attēlot	vaizduõklis, displė̃jus; vaizdúoti, ródyti
Menu	izvēlne	meniù
Address	adrese	ãdresas
Button	poga	klavìšas; mygtùkas

4 How Similar Latvian and Lithuanian ICT Terms Are?

Based just on small sample of the most frequently used ICT terms, we can state the following:

- there are 19 terms with common word-roots (46%), of them 15 (36%) are internacionalisms;
- four terms (10%) are with common Baltic word-roots;
- rather small amount of true anglicisms (5 in Lithuanian, 4 in Latvian; 12% and 10%, respectively) are in use.

By no means, the sample is too small to make statistically confident conclusions about all 5–10 thousands of ICT terms. However, we can cautiously assume that

- commonalities between Latvian and Lithuanian ICT terms are mostly due internationalisms, there are only small proportion of terms with common Baltic word-roots;
- influence of English language in Latvian and Lithuanian ICT terminology is moderate, if not minor;
- deliberately or unawares, Lithuanian terminologists follow the same rules as their Latvian counterparts.

Acknowledgements. The work described in this paper was supported by the project no. AAP2016/B032 “Innovative Information Technologies” at the University of Latvia.

References

1. Kvašytė, R.: Tarp Lietuvos ir Latvijos: lingvistinės paralelės. Mokslinių straipsnių rinkinys. Šiauliai: Šiaulių universiteto leidykla, [Bibliotheca actorum humanitaricorum universitatis Saulensis, knyga nr. 12], 576 p. (2012) (in Lithuanian). ISSN 1822-7257. ISBN 978-609-430-177-3
2. Sabaliauskas, A.: Lietuviu kalbos leksika. Vilnius, Mokslas (1990)
3. Borzovs, J., Ilziņa, I., Skujiņa, V., Vancāne, I.: Sistēmiska latviešu datorterminoloģijas izstrāde._ LZA Vēstis, 55.sēj., 1./2.num., lpp. 83–91 (2001) (in Latvian). https://www.researchgate.net/publication/253153992_Sistemiska_latviesu_datorterminologijas_izstrade
4. Borzovs, J., Ilziņa, I., Keiša, I., Pinnis, M., Vasiļjevs, A.: Terminology localization guidelines for the national scenario. In: Proceedings of International Conference on Language Resources and Evaluation (LREC 2014), Reykjavík, Iceland, May 26–31, pp. 4012–4017 (2014). https://www.researchgate.net/publication/263165203_Terminology_localization_guidelines_for_the_national_scenario
5. EuroTermBank. http://www.eurotermbank.com/default.aspx?lang=lv

Invited Talks

Business Process Analytics: From Insights to Predictions

Marlon Dumas(✉)

University of Tartu, Tartu, Estonia
marlon.dumas@ut.ee

Abstract. Business process analytics is a body of methods for analyzing data generated by the execution of business processes in order to extract insights about weaknesses and improvement opportunities, both at the tactical and operational levels. Tactical process analytics methods (also known as process mining) allow us to understand how a given business process is actually executed, if and how its execution deviates with respect to expected or normative pathways, and what factors contribute to poor process performance or undesirable outcomes. Meantime, operational process analytics methods allow us to monitor ongoing executions of a business process in order to predict future states and undesirable outcomes at runtime (predictive process monitoring). Existing methods in this space allow us to predict for example, which task will be executed next in a case, when, and who will perform it? When will an ongoing case complete? What will be its outcome and how can negative outcomes be avoided. This keynote paper presents a framework for conceptualizing business process analytics methods and applications. The paper and the keynote provide an overview of state-of-art methods and tools in the field and outline open challenges.

Keywords: Business process management
Business process monitoring · Process mining
Predictive process monitoring

1 Introduction

A business process is a *collection of inter-related events, activities, and decision points that involve a number of actors and objects, which collectively lead to an outcome that is of value to at least one customer* [2]. Typically, a business process has a well-defined start and end, which gives rise to a notion of an *instance* of the process, also known as *case*. Such a case is the particular occurrence of business events, execution of activities, and realization of decisions to serve a single request by some customer.

The way in which a business process is designed and performed affects both the *quality of service* that customers perceive and the *efficiency* with which services are delivered. An organization can outperform another organization

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 15–20, 2018.
https://doi.org/10.1007/978-3-319-97571-9_3

offering similar kinds of service, if it has better processes and executes them better. However, the context of business processes in terms of organizational structures, legal regulations, and technological infrastructures, and hence the processes themselves are subject to continuous change. Consequently, to ensure operational excellence, one-off efforts are not sufficient. Rather, there is a need to continuously assess the operation of a process, to identify bottlenecks, frequent defects and their root causes, and other sources of inefficiencies and improvement opportunities.

In this setting, business process analytics is a body of concepts, methods, and tools that allows organizations to understand, analyze, and predict, the performance of their business processes and their conformance with respect to normative behaviors, on the basis of data collected during the execution of said processes. Business process analytics encompasses a wide range of methods for extracting, cleansing, and visualizing business process execution data, for discovering business process models from such data, for analyzing the performance and conformance of business processes both in a post-mortem and in an online manner, and finally, for predicting the future state of the business processes based on their past executions [3].

The common feature of business process analytics techniques is that they allow us to analyze business processes based on their representations in the form of event data and/or business process models. Figure 1 provides a general overview of these techniques and the context in which they are defined. This context is given by a business process as it is supported by an enterprise system, the event data that is recorded during process execution (available as event logs or streams), and the models that capture various aspects of a business process.

2 Event Streams and Event Logs

The left-hand-side of Fig. 1 shows that the starting point for business process analytics is the availability of event streams and/or event logs generated by the execution of business processes.

Modern enterprise software systems keep track of the state of each case at any point in time. To this end, they create a data records, commonly referred to as an *events*, whenever an automated activity is executed, a business event occurs, a worker starts or finishes working on a work item, or a decision is made. Each of these events typically includes additional data, e.g., as it has been provided as a result of executing an activity, whether it was automated (e.g., the response message of a Web service call) or conducted manually (e.g., data inserted by a worker in a Web form). For example, after triggering an automated plausibility check of a loan request, an enterprise software system determines whether to forward the request to a worker for detailed inspection. If so, a worker manually determines the conditions under which a loan is offered. A definition of these conditions is then submitted to the system when completing the respective work item.

While coordinating and performing activities related to the cases of a business process, an enterprise software system produces an *event stream.* Each event

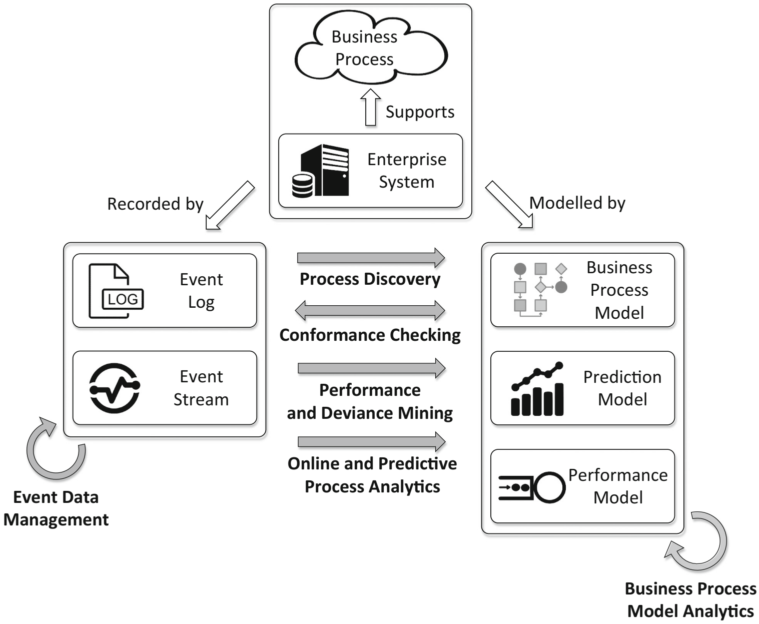


Fig. 1. Overview of business process analytics techniques (taken from [3])

denotes a state change of a case at a particular point in time. Such an event stream may directly be used as input for business process analytics. For instance, using stream processing technology, compliance violations (loans are granted without a required proof of identity of the requester) or performance bottlenecks (final approval of loan offers delays the processing by multiple days) can be detected.

In many cases, business process analytics is done in a post mortem manner for tactical management processes (e.g. to detect bottlenecks, root causes of issues, etc.). To support such tactical use cases, events produced by a business process are recorded and stored in a so-called *event log*, which is analyzed later using a range of techniques. An event log is a collection of events recorded during the execution of (all) cases of a business process during a period of time. Techniques that rely on such event logs for tactical analysis and decision making are commonly known as process mining techniques [1]. These techniques encompass automated process discovery, conformance checking, and performance and deviance mining as discussed below.

2.1 Process Discovery and Conformance Checking

Assuming that we have an event log as described above, a first family of business process analytics techniques allows us to either discover a process model from the event stream/log or to compare a given event stream/log to a given process model. These two latter types of techniques are discussed in the entries on

Automated process discovery [5] and *Conformance checking* [7] respectively (cf. Fig. 1). Automated process discovery techniques take as input an event log and produce a business process model that closely matches the behavior observed in the event log or implied by the traces in the event log. Meanwhile, conformance checking techniques take as input an event log and a process model, and produce as output a list of differences between the process model and the event log. For example, the process model might tell us that after checking the plausibility of a loan application, we must check the credit history. But maybe in the event log, sometimes after the plausibility check, we do not see that the credit history had been assessed. This might be because of an error, or more often than not, it is because of an exception that is not captured in the process model. Some conformance checking techniques take as input an event log and a set of business rules instead of a process model. These techniques check if the event log fulfills the business rules, and if not, they show the violations of these rules.

When applied to complex event logs, the output of automated process discovery and conformance checking techniques are often difficult to understand. One of the main challenges in the field of business process analytics is how to help analysts to navigate through the complexity of event logs generated by complex real-life business processes. Several techniques to tackle these challenges have been proposed in the literature and implemented in both commercial and open-source tools. These techniques can be broadly classified into divide-and-conquer techniques (e.g. trace clustering and modular process discovery) and *declarative process mining*, which rely on summarized representations of a business process in terms of business constraints, as opposed to fully specified (imperative) process models in mainstream notations such as BPMN.

3 Performance and Deviance Analysis

Another core set of business process analytics techniques are focused on business process performance and deviance analysis (cf. Fig. 1). A pre-requisite to be able to analyze process performance is to have a well-defined set of process performance measures. Once we have defined a set of performance measures, it becomes possible to analyze the event log from multiple dimensions (e.g., on a per variant basis, on a per-state basis, etc.) and to aggregate the performance of the process with respect to the selected performance measure. Providing user-friendly visualizations of process performance based on event logs is another challenge in the field of business process analytics.

Naturally, the performance of the process can vary significantly across different groups of cases. For example, it may be that the cycle time 90% of cases of a process is below a given acceptable threshold (e.g., 5 days). However, in some rare cases, the cycle time might considerably exceed this threshold (e.g., 30+ days). *Deviance mining* [4] techniques help us to diagnose the reasons why a subset of the execution of a business process deviate with respect to the expected outcome or with respect to established performance objectives. These techniques take as input two event logs (corresponding to two variants of a process) and produce as output a list of differences. Typically, one of the event logs contains all the cases that end up in a positive outcome according to some criterion, while the other log contains all the cases that end up in a negative outcome. For example, the first log may contain all cases where the customer was satisfied, while the second one contains all cases that led to a complaint. Or the first log may contain all cases where the process completed on time, while the second one contains the delayed cases. Extracting useful summaries of such mismatches in order to help analysts to identify the root causes for performance deviance is an active area of research and development in the field of process analytics.

4 Predictive Process Monitoring

All the techniques for business process analytics mentioned above are primarily designed to analyze the current 'as-is' state of a process in a post mortem setting, in order to support tactical process improvement decisions.

Equally important is the detection of deviations with respect to conformance requirements in real-time, as the process unfolds. This can be achieved by building predictive models from historical event logs, and then using these models to make predictions (in real-time, based on an event stream) about the future state of running cases of the process. The challenge in this setting is how to predict the outcome of a process with high accuracy, but also as early as possible, and how to effectively explain to the user (e.g. process workers), why a given prediction is made and what can be done to alter the course of the process to prevent undesired outcomes. This latter challenge has inspired a stream of research and development known as *predictive process monitoring* [6].

5 Summary and Outlook

Business process analytics is an active field of research and development with several open challenges. A common thread of these challenges is how to make the outputs of business process analytics techniques more interpretable and actionable. For example, it is not enough to identify differences between cases of a business process that end in a negative outcome and those that end in a positive (desired) outcome. Instead, business process analytics techniques need to help business analysts to uncover the root causes for undesired outcomes, and recommend possible mitigations (actionability). This holds both for tactical (offline) process analytics techniques as well as operational (online) ones.

Acknowledgments. This paper borrows from ideas and definitions developed jointly with Matthias Weidlich and captured in the Business Process Analytics entry of Springer's Encyclopedia of Big Data Technologies. The research has been supported by the Estonian Research Council (grant IUT20-55).

References

1. van der Aalst, W.M.P.: Process Mining - Data Science in Action, 2nd edn. Springer, Heidelberg (2016). https://doi.org/10.1007/978-3-662-49851-4
2. Dumas, M., Rosa, M.L., Mendling, J., Reijers, H.A.: Fundamentals of Business Process Management, 2nd edn. Springer, Heidelberg (2018)
3. Dumas, M., Weidlich, M.: Business process analytics. In: Encyclopedia of Big Data Technologies. Springer, Heidelberg (2018)
4. Folino, F., Pontieri, L.: Deviance mining. In: Encyclopedia of Big Data Technologies. Springer, Heidelberg (2018)
5. Leemans, S.: Automated process discovery. In: Encyclopedia of Big Data Technologies. Springer, Heidelberg (2018)
6. Maggi, F.M., Di Francescomarino, C., Dumas, M., Ghidini, C.: Predictive monitoring of business processes. In: Jarke, M., et al. (eds.) CAiSE 2014. LNCS, vol. 8484, pp. 457–472. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-07881-6_31
7. Munoz-Gama, J.: Conformance checking. In: Encyclopedia of Big Data Technologies. Springer, Heidelberg (2018)

Towards the Next-Generation Enterprise Information Systems – Research Opportunities and Challenges

Milan Zdravković(✉)

Faculty of Mechanical Engineering in Niš, University of Niš, Niš, Serbia
milan.zdravkovic@masfak.ni.ac.rs

Abstract. The increase in computing and storage performance achieved in a near past has brought attention of the researchers and practitioners back to different paradigms that could be effectively and efficiently used to solve many societal problems, such as Cloud architectures, Internet of Things (IoT), Artificial Intelligence (AI), Multi-Agent Systems (MAS) and Blockchain. This position paper discusses how all those new technologies, approaches and methodologies affect the landscape of Enterprise Information Systems (EIS). First, it identifies the properties of so-called Next-generation EIS (NG EIS). Then, it proposes the original approach to interoperability of systems, as inherent parent property of EIS. Finally, it presents the framework solution for the implementation problem. The proposed concepts are based on a wide discussion among members of IFAC TC5.3 (IFAC Technical Committee for Enterprise Integration and Networking) committee of the International Federation of Automatic Control.

Keywords: Enterprise information systems · Semantic interoperability
Internet of Things · Multi-agent systems · Systems implementation
Model-driven architecture

1 Introduction

The landscape of Enterprise Information Systems (EIS) has been challenged in the past several years by the trend of introduction of several disrupting technologies, such as Internet of Things (IoT), Artificial Intelligence (AI) and Blockchain, to name just a few. Although each of those technologies are strongly founded in the state of the art in automatic control, networking and statistics since decades ago, the improvements in computational processing power and storage and all pervasive connectivity has made them become practical and useful today, at the massive scale.

According to IBM Marketing Cloud study [1], as much as 90% of all data on the Internet has been created since 2016. Availability and accessibility (through increased use of Application Programming Interfaces – APIs) of big data have caused a paradigm shift not only in consideration of EIS architectures but also in economy at large. Complexity and diversity of capabilities to process known data (functional paradigm) as a primary feature of the traditional EIS becomes even less important than capability to discover, understand

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 21–31, 2018.
https://doi.org/10.1007/978-3-319-97571-9_4

and use new data. This can only be achieved in data centric architectures, where the data model is the primary and permanent asset and it precedes the implementation of any given application. However, data models must be associated with appropriate processing models, which are typically a part of business logic layer in EIS. Consequently, this approach poses the need for having also business logic models.

Unfortunately, only 10% of data on Internet is structured [2], meaning that it's not convenient for use in M2M interactions without some kind of pre-processing. Data mining, text mining and similar technologies, side by side with machine learning algorithms are today increasingly used to make sense of both unstructured and structured data. As variety of different approaches and computational complexity increases, it becomes more and more difficult to embed selected algorithm in the business logic layer of the EIS.

Although widespread use of cloud technologies has helped many enterprises to improve their IT capabilities with minimal implementation risks, cloud architectures bring also issues related to centralization, such as vendor lock-in, data protection issues and application silos. This can be a particularly big problem with the paradigms for which there are almost none feasible alternatives to cloud-based solutions, such as Internet of Things. Still, as the devices in IoT becomes more capable to pre-process and process data, decentralization and use of new peer-to-peer architectures becomes seriously considered choice, even despite some risks, mostly security-related. Those architectures are typically based on Multi-Agent Systems (MAS) and since recently, the Blockchain.

Key benefit from considering MAS architectures is not in decentralization. In fact, it lies in the opportunity to use the power of intelligent agents to have EIS autonomously and automatically represent one enterprise in the rising number of digital collaborative spaces, such as bidding and negotiation platforms, social media, CRM (chat-bots) etc.

While the technological innovation needed for developing EIS that could address the problems above is already there, the challenges arising from implementing and maintaining its complex infrastructure, and creating interoperable ecosystems, are still under researched. These challenges start even at the adoption level, where lack of public policy and regulatory measures is an obstacle, both at macro (achieving full connectivity and interoperability) and micro (facilitating trustworthiness and incentivization) levels.

The risks and uncertainty of sensing EIS systems implementations increase when we consider the technical and organizational efforts and associated change, required by the enterprises. Even the adoption and implementation of conventional EIS is still a big challenge for them. In 2010, the mean Enterprise Resource Planning (ERP) systems' implementation cost was $5.48 million, and the average implementation time-frame was 14.3 months [3]. The failure rate of IT projects remains significant. McKinsey and University of Oxford's research have shown that "on average, large IT projects run 45% over budget and 7% over time, while delivering 56% less value than predicted" [4]. Due to increased complexity and interoperability requirements, it is expected that the failure rates of IoT projects' implementations will be higher. According to IDC, 85% of existing devices worldwide are based on unconnected legacy systems [5].

Future EIS will need to integrate IoT platforms with existing EIS infrastructures to encompass cross-domain sensing and actuating capabilities, thus introducing additional complexity and major risks when considering the implementation. The integration will be considered as tight - great most of existing IoT platforms are only big data aggregators, meaning that additional functionalities will need to be used by the other subsystems - components. Furthermore, IoT platforms are typically driven by models of the trivial complexity; they support very simple data structures and almost no business logic implementation. Finally, IoT systems are today usually managed centrally, which in context of the heterogeneous environment of the IoT ecosystem often means more compromises on openness and greater change management costs. Thus, the problem of the future EIS implementation can be summarized to two simple questions: How to deliver and maintain the required (continuously changing) functionalities? How to deliver in an "open" eco-system of the heterogeneous components, technologies and standards?

This position paper summarizes the ongoing discussion about the Next generation EIS (NG EIS) between the members of IFAC TC5.3 (IFAC Technical Committee for Enterprise Integration and Networking) committee of the International Federation of Automatic Control. The discussion takes place at different venues, mostly International Workshops on Enterprise Integration, Interoperability and Networking (EI2N) and IFAC Symposium on Information Control Problems in Manufacturing (INCOM), but also numerous special issues of the respectable international journals and special tracks at the other conferences, organized by IFAC TC5.3 members. The discussion aims to make a proposal for a high-level architecture of next generation EIS and methodology for its implementation.

2 High-Level Architecture of NG EIS

In the introductory section, several market circumstances, based on the technical developments were shortly considered as important for the further development of EIS. To summarize, those circumstances were identified as:

1. increased computational and storage capability available;
2. sensing and actuating equipment and devices with connectivity (even some minimal storage and processing capabilities) available at low cost;
3. increased availability and accessibility (API) of relevant data;
4. digital collaboration platforms with their APIs facilitate automated representation of the enterprise;
5. significantly increased requirements related to enterprise flexibility in terms of rapid implementation of new data and functions in daily IT operations;
6. architectural patterns based on the centralized management and control of digital assets.

In this section, based on above, the properties of the NG EISs are identified. Then, based on these properties, high-level, abstract architecture is proposed.

2.1 Properties of NG EIS

In the effort which is expected to produce a proposal for the high level architecture of NG EIS, based on the identified circumstances above, the list of relevant properties of the future EISs is compiled [6]. Those features are shortly discussed below.

Omnipresence. Traditional deployment platforms for EISs are enterprise computer infrastructures. Only recently, EISs started to get hosted on private and public clouds. As other different devices, such as sensors and actuators gain processing power, NG EIS must consider them as new deployment platforms. Thus, as new platforms are being embedded in the physical enterprise resources (e.g. transport, manufacturing) the NG EIS will become pervasive, ubiquitous, omnipresent.

Model-driven Architecture. In data-centric economy, enterprise must exhibit the capability to respond and adapt to new data sources as they become available and accessible. Such capability cannot be realized by hard-coded models. In contrast, its precondition is that different models of the business realities, with associated methods and paradigms not only for transformation of these models to executable code, but to their real-time use become important enterprise assets [7]. Hence, the complexity of the contexts in which NG EIS operates (business logic, in terms of traditional systems architecture), will be moved out of its core native environment to the independent, reusable, partially open models' infrastructure.

Openness. Above mentioned models infrastructure will be distributed and collaboratively maintained, tightly integrated with enterprise knowledge management infrastructure, at disposal of all dispersed NG EIS components. Moreover, this infrastructure will embed the different, functionally, logically and geographically distributed knowledge resources, published by the enterprise's partners, suppliers, but sometimes even public, e.g. Open Linked Data.

Dynamic Configurability. The models will be used also to dynamically configure and reconfigure the system in specific circumstances. NG EIS will be capable to search for and automatically use the required distributed computing elements (e.g. services) and models from cloud repositories [8]. Besides functional issues, this will address also non-functional requirements, such as security, performance, storage, etc.

Multiple Identities. NG EISs will represent the enterprise in multitude of collaborative platforms with the specific built identities/profiles, technically implemented as intelligent agents and avatars. Each of the identities will be using formal representation of goals, as well as formal models of the environments in which they act towards the goal fulfilment.

Awareness/Inclusive Sensing. In a long term, NG EIS will exhibit awareness of their environment (including self-awareness) as the property evolved from the system integration paradigm. The awareness assumes also the capability to sense, perceive and understand the messages exchanged with other systems, as well as various multi-modal stimuli from its environment. Aware NG EISs will respond to the recognized need to develop sensing capabilities in an enterprise.

Computational Flexibility. Increased complexity will often pose the need to consider the progressive abandonment of the deterministic approach in business applications. NG EIS will also provide associated computational assets that will allow the system to seamlessly combine deterministic and non-deterministic reasoning in specific circumstances, where the latter also includes machine learning algorithms. The computational flexibility will be achieved by including the abstract representations of those computational assets into overall models infrastructure.

2.2 Abstract Architecture of NG EIS

One NG EIS needs to exhibit the above listed properties while maintaining the core horizontal features, such as security, trust, scalability, integrity and performance. Although the perception of importance and relevance of those features for the sensing EIS has considerably changed, they will not be discussed in this paper.

The list of properties implies that interoperability is foreseen as an inherent property of the NG EIS. It facilitates forming ad-hoc Enterprise Information Ecosystems, spanning the boundaries of the multiple enterprises; it uses the correspondences between the different models and enables their use by the core execution environments; it integrates data, information and knowledge resources and services; it enables a federation of NG EIS functions to its smart objects, agents and avatars; it facilitates ad-hoc communication of the different systems.

The listed properties are used to propose the generic abstract architecture of the NG EIS [6], as illustrated in Fig. 1.

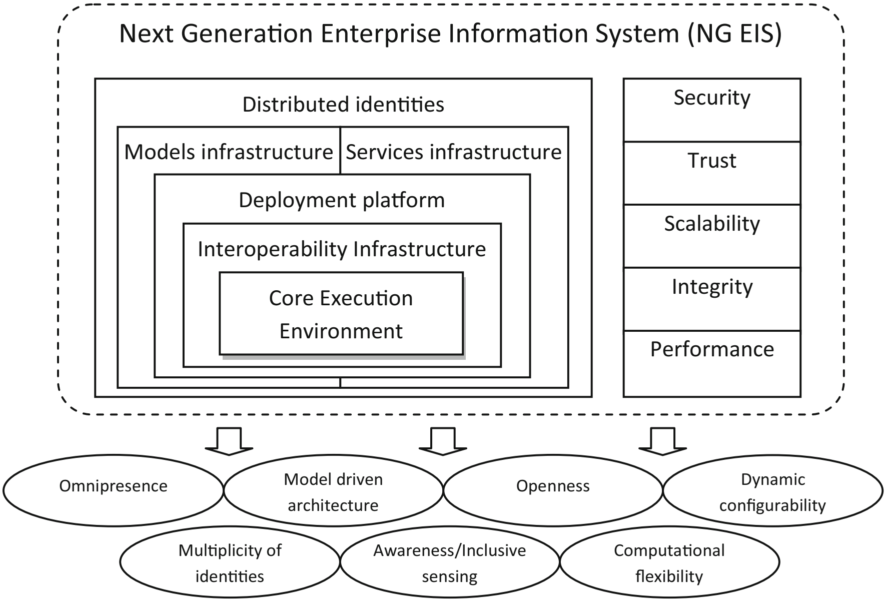

Next Generation Enterprise Information System (NG EIS)
Distributed identities
Models infrastructure
Services infrastructure
Deployment platform
Interoperability Infrastructure
Core Execution Environment
Security
Trust
Scalability
Integrity
Performance
Omnipresence
Model driven architecture
Openness
Dynamic configurability
Multiplicity of identities
Awareness/Inclusive sensing
Computational flexibility

Fig. 1. Abstract architecture of NG EIS

2.3 Semantic Interoperability as Inherent Property of NG EIS

It is argued above that interoperability is *sine qua non* for the NG EIS. With so many protocols for M2M communication on the market (e.g. CoAP, MQTT, XMPP and others), it is sometimes believed that interoperability problem can be reduced to mapping and transformation challenge. Also, it is currently perceived that, there has to be an agreement between interoperating parties on how they will interoperate, before actually interoperation takes place.

The above assumption cannot hold in open EI ecosystems with uncertain heterogeneity, neither interoperability can be reduced to simple translation. Actually, interoperability is often related to the federated approach (ISO/IEC 2382), which implies that a few or, in ideal situation no pre-determined assets for interoperations are assumed. In reality, this means that in order to interoperate with another device, each device must sense, perceive, interpret and understand data, sent from another device and act (operate) upon this understanding. Those capabilities are attributes of semantic interoperability and they are often referred to as the building blocks for intelligent behaviour.

In the contemporary research of semantic interoperability, the process of understanding is reduced to the process of reasoning about the transmitted concepts, in which the meaning is assigned to transmitted messages, in a real time. However, there exist opinions that the more complex, actually anthropomorphic perspective to semantic interoperability could provide better results. In such a perspective, the semantic interoperability, so far considered as unidirectional property of a pair of the systems [10] that interoperate, is now replaced with an assumption that it is in fact an inherent property of a single system, realizing its capabilities to sense, perceive its environment, make an intelligent decision on the response to a perceived meaning of the stimuli and articulate this response [11] (see Fig. 2). The assumption is based on analogies with the human communication process, which is today considered as interplay of 4 physiological and psychological groups of processes: sensation (physiological), perception, cognition and articulation [12].

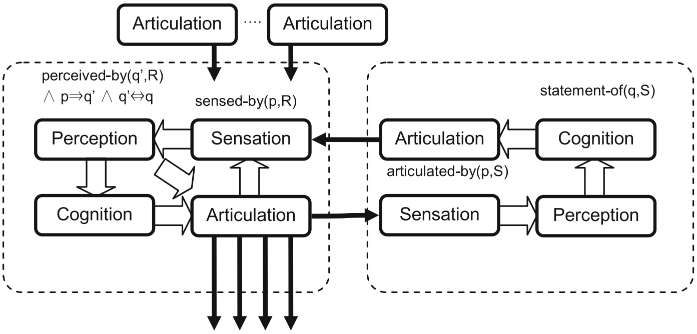


Fig. 2. Anthropomorphic perspective to semantic interoperability

Consequently, the interoperability property of one NG EIS is considered as constituted by the corresponding attributes of awareness, perceptivity, intelligence and extroversion. These attributes can be further decomposed [13], as it is further elaborated.

For example, there exist a need to separately consider the self-awareness and environmental awareness of NG EIS. While the latter is important for realizing the omnipresence feature, the former is relevant for maintaining NG EIS's multiple identities. The omnipresence of a NG EIS extends the conventional domains of interest of an enterprise (e.g. typical channels for detecting new business opportunities). Hence, now, one has to consider not only the functional environmental awareness of NG EIS, but also a universal awareness concerning observations of any stimuli, even from unknown and unanticipated sources. Thus, it becomes important for the system to achieve the capability to perceive any stimulus - be it multi-modal, multi-dimensional, discrete or continuous.

Perceptivity is a capability of a NG EIS to assign a meaning to the observation from its environment or from within itself. When considering the awareness capabilities, mentioned above, we can distinguish between the perceptivity related to perceiving the sensor data and the perceptivity related to assigning a meaning to an incoming message. Both are based on the access to a wide range of perceptual theories or sets, consisting of different ontologies representing the domain knowledge, but also motivational aspects of communication, e.g. different problem or application ontologies.

Then, based on the perception, the NG EIS should be able to decide on the consequent action. This decision is a result of a cognitive process, which consists of identification, analysis and synthesis of the possible actions to perform in response to the "understood" observation. Consequently, we can distinguish between observational and communicative perceptivity. The intelligence also encompasses assertion, storing and acquisition of the behaviour patterns, based on the post-agreements on the purposefulness of the performed actions.

The last desired attribute of a NG EIS, extroversion, is related to its willingness and capability to articulate the above action/s and it demonstrates the enterprise's business motivation and/or a concern about its physical and social environment.

3 Implementation Problem Revisited

The conceptual solution to NG EIS implementation problem is sought in the different domains, based on the following line of thought.

First, it is clear that enterprises need to have a wide understanding of the inner workings and impact of the NG EIS and IoT ecosystem in which it operates. This understanding and a shared agreement is established by using models. Second, models are developed as a result of the complex communication between different stakeholders. Such communication needs framework which ensures that modelled artefacts implement system requirements; this framework is typically established by using Requirements Engineering practices and tools. Third, the models specify some important technical decisions made after the requirements analysis. One of the most important ones is the system architecture. Multi-agent systems can decentralize NG EIS and enable devices to make decisions locally. Fourth, as previously noted interoperability is one of the greatest challenges in making a complex NG EIS a reality. Though it may be considered also as a system requirement, interoperability is discussed separately because its impact spans multiple domains; it is a core feature of the IoT ecosystems. Both system requirements and architectural concepts must acknowledge that by taking into account interoperability issue in their formal definitions. Fifth, very complex change made by the implemented IoT systems imposes the need to take into account maturity models and associated verification and validation processes. Special case of evaluation must take place in assessment of interoperability, as the most critical requirement for the open IoT ecosystems. Finally, this openness implies a strong need to take into account different societal, policy and legal aspects in their implementation. The above domains are interrelated and they form the proposed Domain framework for addressing IoT implementation problem [14] (see Fig. 3).

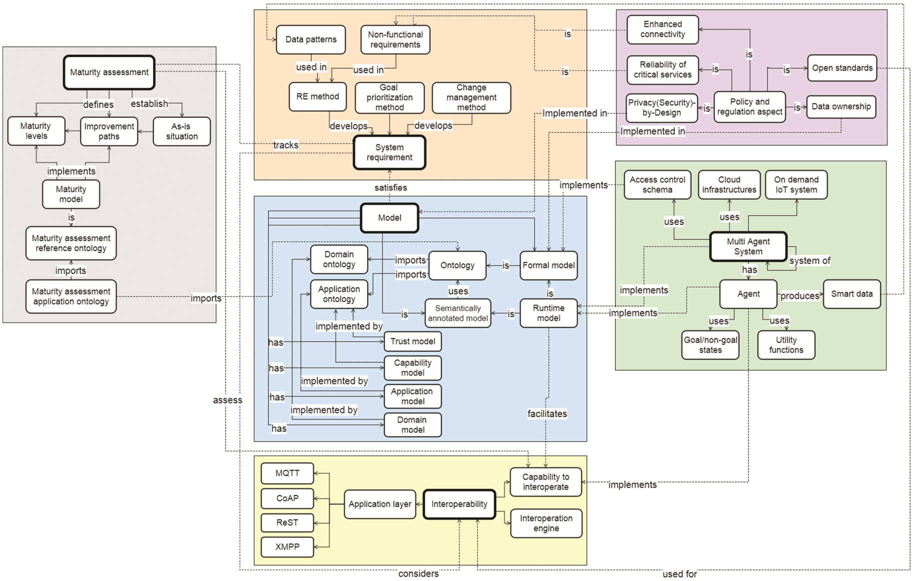


Fig. 3. A domain framework for addressing NG EIS implementation problem

The framework identifies 5 key domain factors for the implementation: System requirements, Multi-agent systems, Interoperability, Maturity assessment and Policy and regulations aspects which affect all former elements. In the reminder of this section, the relationships between key domain elements, dependencies and cross-domain concepts, as illustrated in Fig. 3 are described in detail.

3.1 Effect of Policy and Regulatory Aspects to the Domain Framework Elements

Enhanced connectivity, namely new M2M services will affect the structure of non-functional requirements collection and associated meta-data. Access control schemas, used in MAS, must be unified [15] and considered at modelling level. They will be formally expressed, while continuously taking into account data ownership issues, including licenses of data use in distributed environment of IoT ecosystem.

In the development of model frameworks both domain experts and system architects must follow privacy-by-design and security-by-design policies. Reliability of critical services, such as healthcare, safety and security will be considered as non-functional requirements of highest priority and implemented in traffic prioritization policies. Finally, capability to interoperate will be based on the open interoperability standards.

3.2 Core Technical Structure of the Domain Framework

The backbone of the framework consists of: (1) system requirements, (2) modelling constructs that satisfy them and (3) agents which implement the models. The centrepoint of this backbone is the model. It is either semantically annotated or formal model (so it facilitates inferring the meaning of the data coming from different sources); it is interpreted at runtime, by the implementation agent in MAS (core-execution environment).

Besides representing the agent environment, formal models are used to define goal and non-goal states, to be reached by the goal-based agents and utility functions to be used by the utility-based agents to measure how desirable perceived state is. In satisfying the system requirements and making sense of acquired data, besides defining structural (including data structures and restrictions, explicitly defined in domain ontology) and behavioural aspects (explicitly defined in application ontology), model also considers innovative views to the ecosystem, such as capabilities of its artefacts, maturity and trust.

The capability model is a cross-domain concept, which is proposed due to a need to abstract the heterogeneity of devices and to formally define the capability of one device or agent to interoperate with another (self-awareness property). It is formally implemented as extension of W3C SSN ontology [16] in application ontology.

The trust model is introduced by the need to facilitate the agent's capability to acquire data from relevant, reliable and trustful sources. The trust model will also define formal requirements for models' validation. Trust ontologies have started to emerge, even with specializations in IoT domain [17]; most of the current models have been built on the O'Hara's formal trust model of trustworthiness [18].

The maturity model is a cross-domain concept, and it is used to formally define improvement path and maturity levels as agent goals. The maturity model is instantiation of the Maturity assessment application ontology, which specialize the upper-level Maturity assessment reference ontology (for example, based on [19]) in the domain (formally described by the domain ontology) and application context.

The application model is meta-model which is used to instantiate behavioural aspects of the IoT scenarios in the eco-system, such as services (CRUD, processing, visualization and others) and their orchestration (business processes), access control schema, user interfaces (if any), etc.

4 Conclusion

This position paper summarizes the discussion about the research challenges and opportunities about the future generation EIS among individual experts in the IFAC TC5.3 (IFAC Technical Committee for Enterprise Integration and Networking) committee of the International Federation of Automatic Control. As such, it is not the committee's manifest nor it reflects the common position towards the subject by all its members. Instead, it presents several conceptual frameworks by the group of members, presented in some multi-author papers, published in respected international journals. Those conceptual frameworks are:

- high-level architecture of NG EIS, addressing its identified critical properties;
- new formal definition of semantic interoperability, considering it key capability of NG EIS;
- domain framework for NG EIS implementation.

All of the above contributions should be considered as blueprints for ongoing and future research within and outside the group. Although there are no strong feasibility evidences and real-world validation of the proposed frameworks, the scientific method is followed to a certain extent; it encompassed the detailed analysis and synthesis of number of existing works, as well as case studies (presented in the cited papers) in which the future use of frameworks was described in detail.

References

1. IBM Marketing Cloud. 10 key marketing trends for 2017 and ideas for exceeding customer expectations. https://public.dhe.ibm.com/common/ssi/ecm/wr/en/wrl12345usen/watson-customer-engagement-watson-marketing-wr-other-papers-and-reports-wrl12345usen-20170719.pdf
2. Schubmehl, D.: Unlocking the hidden value of information. IDC Community (2014). https://idc-community.com/groups/it_agenda/bigdataanalytics/unlocking_the_hidden_value_of_information
3. Galy, E., Sauceda, M.J.: Post-implementation practices of ERP systems and their relationship to financial performance. Inf. Manag. **51**, 310–319 (2014)
4. McKinsey Report. Delivering Large-scale IT Projects on Time, on Budget, and on Value (2012)
5. IDC. Business Strategy: The Coming of Age of the "Internet of Things" in Government. IDC, April 2013
6. Panetto, H., Zdravković, M., Jardim-Goncalves, R., Romero, D., Cecil, J., Mezgar, I.: New perspectives for the future interoperable and sustainable enterprise systems. Comput. Ind. **79**, 47–63 (2016)
7. Agostinho, C., Černý, J., Jardim-Goncalves, R.: MDA-based interoperability establishment using language independent information models. In: 4th International IFIP Working Conference on Enterprise Interoperability (IWEI 2012), Harbin, China, pp. 146–160. Springer, Heidelberg (2012)

8. Ducq, Y., Chen, D., Alix, T.: Principles of servitization and definition of an architecture for model driven service system engineering”. In: In: 4th International IFIP Working Conference on Enterprise Interoperability (IWEI 2012), Harbin, China, pp. 117–128. Springer, Heidelberg (2012)
9. Jardim-Goncalves, R., Grilo, A., Popplewell, K.: Sustainable interoperability: the future of internet based industrial enterprises. Comput. Ind. **63**(8), 731–738 (2012)
10. Sowa, J.: Knowledge Representation: Logical, Philosophical, and Computational Foundations. Brooks/Cole Publishing Co., Pacific Grove, CA (2000)
11. Zdravković, M., Luis-Ferreira, F., Jardim-Goncalves, R., Trajanović, M.: On the formal definition of the systems’ interoperability capability: an anthropomorphic approach. Enterp. Inf. Syst. **17**(3), 389–413 (2015)
12. Littlejohn, S.W., Foss, K.A.: Theories of Human Communication. Waveland Press Inc., Long Grove (2010)
13. Zdravković, M., Noran, O., Trajanović, M.: Towards sensing information systems. In: Proceedings of the 23rd International Conference on Information Systems Development, Varaždin, Croatia, 2–4 September 2014
14. Zdravković, M., Zdravković, J., Aubry, A., Moalla, N., Guedria, W., Sarraipa, J.: Domain framework for implementation of open IoT ecosystems. Int. J. Prod. Res. (2017). https://doi.org/10.1080/00207543.2017.1385870. (in press)
15. Rivera, D., Cruz-Piris, L., Lopez-Civera, G., de la Hoz, E., Marsa-Maestre, I.: Applying an unified access control for IoT-based intelligent agent systems. In: IEEE 8th International Conference on Service-Oriented Computing and Applications (SOCA), Rome, Italy October 19–21, 2015, pp. 247–251. IEEE (2015)
16. Compton, M., et al.: The SSN ontology of the W3C semantic sensor network incubator group. Web Seman. Sci. Serv. Agents World Wide Web **17**, 25–32 (2012)
17. Taherian, M., Jalili, R., Amini, M.: PTO: a trust ontology for pervasive environments. In: 22nd International Conference on Advanced Information Networking and Applications - Workshops, 2008. AINAW 2008. 22nd International Conference on Advanced Information Networking and Applications - Workshops, Okinawa, Japan, March 25–28, pp. 301–306. IEEE (2008)
18. O’Hara, K.: A General Definition of Trust. Technical report, University of Southampton (2012)
19. Guédria, W., Naudet, Y., Chen, D.: Maturity model for enterprise interoperability. Enterp. Inf. Syst. **9**(1), 1–28 (2015)

Languages of Baltic Countries in Digital Age

Inguna Skadiņa[1,2,3](✉)

[1] University of Latvia (UL), Raiņa bulv. 19, Riga, Latvia
inguna.skadina@lu.lv
[2] Institute of Mathematics and Computer Science UL, Riga, Latvia
[3] Tilde, Riga, Latvia

Abstract. Today, when we are surrounded by intelligent digital devices – computers, tablets and mobile phones, we expect communication with these devices in a natural language. Moreover, such communication needs to be in our native language. We also expect that language technologies will not only assist us in everyday tasks, but also will help to overcome problems caused by language barriers. This keynote will focus on language resources and tools that facilitate use of languages of three Baltic countries – Estonian, Latvian and Lithuanian -in digital means (computers, tablets, mobile phones), allow to minimize language barriers, facilitate social inclusion, and support more natural human-computer interaction, thus making digital services more "human". Current situation, technological challenges and most important achievements in language technologies that help to narrow technological gap, facilitates use of natural language for interaction between computer and human, and minimize threat of digital extinction will be presented.

Keywords: Language resources · Natural language processing
Languages of Baltic countries · Under-resourced languages · Corpora
Speech processing

1 Introduction

Natural language is indispensable mean for communication between humans of all nationalities regardless number of speakers and history of the nation. Languages of Baltic countries - Estonian, Latvian and Lithuanian - are among languages with rather small number of speakers. The number of speakers of particular language is one of the factors that influence amount and availability of digital (and also printed) language resources. Languages of Baltic countries are often mentioned among under-resourced or low-resourced languages. Although there is no precise definition what under-resourced language means, usually it is understood as a language that is insufficiently (in size and quality) represented in a digital form. Lack of language resources (different texts, dictionaries, transcribed audio and video materials, etc.) in a digital form in a turn influence (in fact limits) development of the language technology solutions.

At the beginning of this decade, the META-NET Network of Excellence forging the Multilingual Europe Technology Alliance[1] conducted study on 30 European languages

[1] http://www.meta-net.eu.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 32–40, 2018.
https://doi.org/10.1007/978-3-319-97571-9_5

and the level of support these languages receive through language technologies. This survey was published in a book series, describing state of art for each language. The national language technology landscapes presented in Whitepapers contain general facts about each language and its particularities and describe recent developments in the language technology and core application areas of language and speech technology. The language technology landscape of Baltic languages was described in the following volumes: “Estonian Language in Digital Age” [1], “Latvian Language in Digital Age” [2] and “Lithuanian Language in Digital age” [3].

The language Whitepapers also present a cross-language comparison ranking the respective language within four key areas: machine translation, speech processing, text analysis, and resources. As it is demonstrated in Table 1, the support for Latvian and Lithuanian in all four key areas, when compared to other European languages, was assessed as weak. For Estonian, support for speech and text resources and speech processing was assessed as fragmentary, while machine translation and text analysis support was assessed as weak. It needs to be mentioned that among 30 European languages only four under-resourced languages – Icelandic, Latvian, Lithuanian and Maltese - were assessed as ‘weak support’ in all four key areas.

Table 1. Language technology support for languages of Baltic countries presented in META–NET Whitepapers

	Speech and text resources	Text analysis	Machine translation	Speech processing
Estonian	fragmentary	weak support	weak support	fragmentary
Latvian	weak support	weak support	weak support	weak support
Lithuanian	weak support	weak support	weak support	weak support

The Language Whitepapers have identified serious gaps in basic language resources and technologies (LRT) for all three languages of Baltic States. This paper highlights some most important achievements made to narrow the so-called technological gap in language technology field in three Baltic countries during last five years. We demonstrate that although there is still a gap between well represented English language and under-resourced languages of Europe, the language technologies in Baltic countries have made big step further to overcome this gap and thus making communication of users with computers easier and more attractive.

2 Initiatives and Projects

Languages always have been among priorities of European Union multilingualism policy supporting Europe’s rich linguistic diversity. Necessity of language technology support in digital environment has been acknowledged by both – international and national – institutions. Recent Horizon 2020 ICT call among other priorities has highlighted role of language technologies by targeted topic of multilingual next generation internet, aiming at support for technology-enabled multilingualism for an inclusive Digital Single Market[2].

The fundamental support for languages in a digital environment is provided through research infrastructures – CLARIN and ELEXIS. CLARIN – European research infrastructure for language resources and technology – started "from the vision that all digital language resources and tools from all over Europe and beyond are accessible through a single sign-on online environment for the support of researchers[3]". Today CLARIN consortium (20 members and 2 observers) provides easy and sustainable access to digital language data in many languages [4]. ELEXIS[4] – European Lexicographic Infrastructure – is the Horizon 2020 project aiming at creation of a sustainable infrastructure which enables efficient access to high quality lexical data and helps to bridge the gap between different scholarly communities working on lexicographic resources. An approach driven by practical needs has been taken by ELRC[5] – European language resource coordination - initiative. The ELRC network supports collection and maintenance of the language resources in official languages of the EU with aim to help to improve the quality, coverage and performance of automated translation solutions of Connecting European Facility (CEF) digital services.

In case of under-resourced languages, which usually are represented by rather small number of speakers, support from government is crucial for sustainable research and development and long-term survival of these languages. Such support has been provided in Estonia through National Programmes for Estonian Language Technology (NPELT)[6] since 2006. The programme aims to provide language technology means that enables successful operation of the Estonian language in ICT-based world. The outcome of these technological programs – language resources and tools - are freely available to everybody through the website of Center of Estonian Language Resources[7].

In Lithuania, similarly to Estonia, since 2012 research and development in a field of human language technologies is funded through national programs for Lithuanian language support in information society. In 2013 State Commission of the Lithuanian language issued "Guidelines for Lithuanian Language Technologies development 2014–2020" where machine translation, speech analysis, dialogue systems, automatic summarization, semantic technologies, advanced text analysis, compilation of language resources, and others, are defined as priorities. Two national infrastructures - *raštija.lt* (Integrated Information System of the Lithuanian language and language resources) and LKSSAIS (Information system for syntactical and semantical analysis of the Lithuanian language)[8] – support access to tools and resources created through national programs [5]. Five new projects to support Lithuanian language in information society were launched in 2018. These projects aims at development of syntactic and semantic analyzers (SEMANTICS 2), Lithuanian language speech services (LIEPA 2) and

[2] http://ec.europa.eu/research/participants/portal/desktop/en/opportunities/h2020/topics/ict-29-2018.html.

[3] https://www.clarin.eu/content/clarin-in-a-nutshell.

[4] https://elex.is/.

[5] http://www.lr-coordination.eu/.

[6] https://www.keeletehnoloogia.ee.

[7] https://keeleressursid.ee/en/.

[8] http://semantika.lt/.

machine translation systems and localization services, as well as, support creation of information systems for integrated Lithuanian language resources (RAŠTIJA 2) and Lithuanian Language Resources (E.kalba)[9].

The importance of the language technologies for the long-term survival of Latvian has been recognized in the State Language Policy Guidelines for 2015–2020. Research in language technologies has been supported by the State Research Programmes, EU Structural Funds Programmes, and grants from the Latvian Science Council, EU FP7 and Horizon 2020 Programmes. However, there is no language technology program in Latvia. As result research and development activities in human language technologies are fragmented and mostly insufficiently supported. Currently two large-scale research and development projects support creation of missing language resources and development of modern language technology solutions. The project "Full Stack of Language Resources for Natural Language Understanding and Generation in Latvian" aims to create a complex, multi-layered set of essential Latvian language resources (corpora, treebanks, lexicons, etc.) and to demonstrate the potential of these advanced linguistic resources though creation of an innovative NLU and NLG technology [6]. The project "Neural Network Modelling for Inflected Natural Languages" aims to research novel models for applying neural network technologies for core language technology tasks – written language processing, speech processing – and advanced applications – machine translation and human-computer interaction.

3 Language Resources

Usually life of the natural language in digital means starts with a text – digital archives, digitized books, etc. These texts serve not only for general public, but also for research and development of language technology solutions. Today almost every language has corpus that represents particular language, e.g., in our countries they include Balanced Corpus of Modern Latvian [7], Corpus of Contemporary Lithuanian Language DLTK [8], Estonian Reference Corpus [9], and many others.

While text corpora have been developed for all three languages for several decades, more complex linguistic resources, such as syntactically annotated text corpora or treebanks have been created only recently [11–13]. Today all three languages are among 60 languages represented in Universal Dependencies Framework[10].

Besides national corpora and multilingual parallel corpora created from translations, the Latvian-Lithuanian parallel corpus is the first corpus that contains many (more than 56%) texts that are originally written in one of these languages [10].

Although there are many lexical databases available, *tezaurs.lv* is the largest open lexical database for Latvian, currently it contains 295 760 lexical entries, that are compiled from more than 280 sources. It is popular not only among researchers, but also widely used by general public – journalists, students and many others [14]. Each day there are more than 2000 requests. *Tezaurs.lv* aims to be the central computational

[9] https://www.cpva.lt/lt/veikla/paramos-administravimas/es-fondu-investicijos-q6t3/finansuojamos-sritys/informacine_visuomene/lietuviu-kalba-informacinese-technologijose.html.

[10] http://universaldependencies.org/.

lexicon for Latvian, bringing together all Latvian words and frequently used multi-word units and allowing for the integration of other LT resources and tools. The dictionary is enriched with phonetic, morphological, semantic and other annotations and enhanced with language processing tools allowing generation of inflectional forms and selection of corpus examples on the fly. It is available also as an API for the integration into third-party applications.

4 Technologies and Tools

Last few years have been challenging not only for language technology developers, but also for many other fields of computer science, as artificial intelligence, namely deep machine learning, has become popular again. It has great impact on almost every area of language technology, but especially on machine translation, human-computer interaction and speech recognition.

4.1 Best Machine Translation for Complex Less Resourced Languages

Machine translation (MT) was one of the areas that was mentioned in Whitepapers as insufficiently supported for all three languages of Baltic countries five years ago. This situation has changed dramatically recently. English-Latvian and Latvian-English machine translation solutions developed for annual MT system competition WMT, was recognized as the best, wining not only solutions developed by well known research teams, but also *Goggle* and *Microsoft* [15]. The neural machine translation systems developed in Latvia also have been in use in presidency countries of the European Union – Estonia and Bulgaria [16].

4.2 Speech Technologies

Research and development on speech technologies have been known as success of Estonia for a long time. Some most recent achievements include real time speech recognition, content search in audio and emotional speech synthesis (e.g. [17–19]). Latvian and Lithuanian for many years were not so well represented by speech technologies. In Latvia the situation changed when transcribed corpus of spoken Latvian was created [20]. Although the corpus is rather small – it contains only 100 h of transcribed speech, it was good starting point for development of several speech recognition systems for Latvian [21, 22]. Similarly, in Lithuania work on speech recognition started with corpus Liepa[11]. Recently speech recognitions solutions for Latvian and Lithuanian have reached state of art recognition quality and are much better as systems developed by global companies [23].

[11] https://www.raštija.lt/liepa.

4.3 Natural Language Understanding and Generation

Abstract Meaning Representation, AMR [24] is a recent representation of whole-sentence semantic analysis, which has gained great popularity in natural language understanding. Along with English, AMR is being adapted and approbated for other languages, e.g. French, Spanish, Czech, etc. Currently AMR is being tested also for Latvian and, thus, Latvian might be the first one among under-resourced languages for which this approach is approbated.

Researchers form Latvia have successfully participated in international competitions related to natural language understanding (NLU) and generation (NLG) tasks: at SemEval-2016 the top result was achieved in the task on Meaning Representation Parsing [25], while at SemEval-2017 - the top result in the subtask on AMR-to-English Generation [26]. These results give confidence that it is worth to develop further the combined machine learning and grammar-based approach for NLU and NLG. Moreover, they demonstrate that AMR, complemented by FrameNet, Universal Dependencies, Grammatical Framework and other state-of-the-art syntactic and semantic representations, is emerging as a powerful interlingua for cross-lingual applications.

4.4 Human-Computer Interaction

New technologies rise new expectations. Among them is a myth that computers can replace humans not only in simple mechanical tasks, but also in more complicated tasks, as it has been demonstrated by *Amelia*[12] and some other virtual employees. Although natural language understanding is still unsolved problem, human-computer interaction in Baltic countries have been studied for many years (e.g. [27, 28]). Today virtual assistants that can communicate not only in English, but also in Latvian or Lithuanian becomes reality [29]. Human-computer interaction through virtual assistants makes communication with computer more natural, as includes not only text, but also speech, images and even emotions. Virtual assistants can help in libraries, be a guide or teach Lithuanian to Latvians, or multiplication table to children. Bilingual (Lithuanian and English) virtual assistant already serves at Migration Department at the Ministry of the Interior in Lithuania[13]. Very soon Latvian speaking virtual assistant will start work at the Register of Enterprises of the Republic of Latvia.

5 Conclusion

Although languages of Baltic countries – Estonia, Latvia and Lithuania - are represented by rather small number of speakers and often are called under-resourced, all three languages are represented in digital world not only by digital libraries of texts and language resources (corpora, lexicons, etc.), but also by fundamental language technologies, such as spelling checkers, morphological analyzers, taggers and parsers. Where it concerns more advanced technologies (even such, which are usually considered as

[12] https://www.ipsoft.com/amelia/.

[13] http://www.migracija.lt/index.php?2044534709.

basic), the situation differs from language to language – all languages has support for machine translation, speech synthesis and recognition, while solutions that involve natural language understanding are not so developed. Strong national support is necessary for further research and development for all three official languages of Baltic States to support their life in IT world and avoid digital extinction.

Acknowledgements. The research was developed under the University of Latvia contract no. AAP2016/B032 “Innovative information technologies”.

References

1. Liin, K., Muischnek, K., Müürisep, K., Vider, K.: Eesti keel digiajastul – the Estonian Language in the Digital Age. Springer, New York (2012)
2. Skadiņa, I., Veisbergs, A., Vasiļjevs, A., Gornostaja, T., Keiša, I., Rudzīte, A.: Latviešu valoda digitālajā laikmetā – The Latvian Language in the Digital Age. Springer, New York (2012)
3. Vaišnienė, D., Zabarskaitė, J. Lietuvių kalba skaitmeniniame amžiuje – The Lithuanian Language in the Digital Age. Springer, New York (2012)
4. De Jong, F., Maegaard, B., De Smedt K., Fišer, D. and Van Uytvanck, D.: CLARIN: towards FAIR and responsible data science using language resources. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation – LREC 2018, pp. 3259–3264. European Language Resources Association (ELRA), Miyazaki, Japan (2018)
5. Utka, A., Amilevičius, D., Krilavičius, T., Vitkutė-Adžgauskienė, D. Overview of the development of language resources and technologies in Lithuania (2012–2015). In: Human Language Technologies – The Baltic Perspective, pp. 12–19. IOS Press (2016)
6. Gruzitis, N., Pretkalnina, L., Saulite, B., Rituma, L., Nespore-Berzkalne, G., Znotins, A., Paikens, P.: Creation of a balanced state-of-the-art multilayer corpus for NLU. In: Proceedings of the Eleventh International Conference on Language Resources and Evaluation - LREC 2018, pp. 4506–3264. European Language Resources Association (ELRA), Miyazaki, Japan (2018)
7. Levane-Petrova, K.: Līdzsvarots mūsdienu latviešu valodas tekstu korpuss un tā tekstu atlases kritēriji (The balanced corpus of modern Latvian and the text selection criteria). Baltistica **8**, 89–98 (2012)
8. Rimkutė, E., Kovalevskaitė, J., Melninkaitė, V., Utka, A., Vitkutė-Adžgauskienė, D.: Corpus of contemporary Lithuanian language – the standardised way. In: Human Language Technologies – The Baltic Perspective: Proceedings of the Fourth International Conference Baltic HLT, pp. 154–160. IOS Press (2010)
9. Kaalep, H.J., Muischnek, K., Uiboaed, K., Veskis, K.: The Estonian reference corpus: its composition and morphology-aware user interface. In: Human Language Technologies – The Baltic Perspective: Proceedings of the Fourth International Conference Baltic HLT, pp. 143–146. IOS Press (2010)
10. Utka, A., Levane-Petrova, K., Bielinskiene, A., Kovalevskaite, J., Rimkute, E., Vevere, D.: Lithuanian-Latvian-Lithuanian parallel corpus. In: Human Language Technologies – The Baltic Perspective, pp. 260–264. IOS Press (2012)
11. Muischnek, K., Müürisep, K., Puolakainen, T.: Dependency parsing of Estonian: statistical and rule-based approaches. In: Human Language Technologies – The Baltic Perspective, pp. 111–118. IOS Press (2014)
12. Pretkalnina, L., Rituma, L., Saulite, B.: Universal dependency treebank for Latvian: a pilot. In: Human Language Technologies – The Baltic Perspective, pp. 136–143. IOS Press (2016)

13. Bielinskienė, A., Boizou, L., Kovalevskaitė, J., Rimkutė, E.: Lithuanian dependency treebank ALKSNIS. In: Human Language Technologies – The Baltic Perspective, pp. 107–114. IOS Press (2016)
14. Spektors, A., Auzina, I., Dargis, R., Gruzitis, N., Paikens, P., Pretkalnina, L., Rituma, L., Saulite, B.: Tezaurs.lv: the largest open lexical database for Latvian. In: Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC), pp. 2568–2571 (2016)
15. Pinnis, M., Krišlauks, R., Miks, T., Deksne, D., Šics, V.: Tilde's machine translation systems for WMT 2017. In: Proceedings of the Second Conference on Machine Translation, Shared Task Papers, vol. 2, pp. 374–381 (2017)
16. Pinnis, M., Kalniņš, R.: Developing a neural machine translation service for the 2017–2018 European Union Presidency. In: Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (AMTA 2018), MT Users, Boston, vol. 2, pp. 72–83 (2018)
17. Kurimo, K., Enarvi, S., Tilk, O., Varjokallio, M., Mansikkaniemi, A., Alumäe, T. Modeling under-resourced languages for speech recognition. In: Language Resources and Evaluation, pp. 961–987. Springer, Netherlands (2017)
18. Alumäe, T.: Full-duplex speech-to-text system for Estonian. In: Human Language Technologies – The Baltic Perspective, Baltic HLT 2014, Kaunas, Lithuania. IOS Press (2014)
19. Paats, A., Alumäe, T., Meister, E., Fridolin, I. Evaluation of automatic speech recognition prototype for Estonian language in radiology domain: a pilot study. In: Mindedal, H., Persson, M. (eds.) 16th Nordic-Baltic Conference on Biomedical Engineering. IFMBE Proceedings, vol. 48, pp. 96–99. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-12967-9_26
20. Pinnis, M., Auziņa, I., Goba, K.: Designing the Latvian speech recognition corpus. In: Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2014), pp. 1547–1553 (2014)
21. Salimbajevs, A.: Towards the first dictation system for Latvian Language. In: Frontiers in Artificial Intelligence and Applications, Human Language Technologies – The Baltic Perspective, vol. 289, pp. 66–73 (2016)
22. Znotins, A., Polis, K., Dargis, R.: Media monitoring system for Latvian radio and TV broadcasts. In: Proceedings of the 16th Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 732–733 (2015)
23. Salimbajevs, A.: Creating Lithuanian and Latvian speech corpora from inaccurately annotated web data. Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018), pp. 2871–2875 (2018)
24. Banarescu, L., Bonial, C., Cai, S., Georgescu, M., Griffitt, K., Hermjakob, U., Knight, K., Koehn, P., Palmer, M., Schneider, N.: Abstract meaning representation for sembanking. In: Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse (2013)
25. Barzdins, G., Gosko, D.: RIGA at SemEval-2016 Task 8: Impact of smatch extensions and character-level neural translation on AMR parsing accuracy. In: Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval) (2016)
26. Gruzitis, N., Gosko, D., Barzdins, G.: RIGOTRIO at SemEval-2017 Task 9: combining machine learning and grammar engineering for AMR parsing and generation. In: Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval), pp. 924–928 (2017)
27. Koit, M.: Modelling human-computer interaction. In: SCAI, p. 276 (1997)

28. Koit, M.: Modelling attitudes of dialogue participants - reasoning and communicative space. In: Proceedings of the 10th International Conference on Agents and Artificial Intelligence, pp. 581–588 (2018)
29. Vasiljevs, A., Skadina, I., Deksne, D., Kalis, M., Vira, I.: Application of virtual agents for delivery of information services. In: New Challenges of Economic and Business Development – 2017, pp. 667–678 (2017)

Information Systems Engineering

Towards the Trust Model for Industry 4.0

Marina Harlamova and Marite Kirikova(✉)

Riga Technical University, 1 Kalku, Riga 1658, Latvia
marina.harlamova@edu.rtu.lv, marite.kirikova@rtu.lv

Abstract. In highly networked systems, such as Industry 4.0, it is essential to take care that only trustworthy elements participate in the network, otherwise the security of the system might be compromised and its functionality negatively influenced. Therefore it is important to identify whether the nodes in the network can be trusted by other elements of the system. There are different approaches for trust evaluation available in a variety of domains. However, the Industry 4.0 involves both human and artificial participants and imposes human-human, artifact-artifact; and human-artifact relationships in the system. This requires comparable interpretation and representation of trust in several areas. For this purpose the paper discusses trust interpretations and proposes trust models and trust dimensions in three areas relevant to Industry 4.0, namely, in the area of human-human interaction, in the area of human interaction with IT solutions, and in ad-hoc distributed sensing systems.

Keywords: Trust · Trust dimension · Trust model · Industry 4.0

1 Introduction

Industry 4.0 [1] is a concept of automation and data exchange in manufacturing environments that combines advances of cyber-physical systems (CPSs) [2], the Internet of Things (IoT), cloud computing and cognitive computing. In Industry 4.0 context, components of CPSs would typically be industrial machines, automated guided vehicles, robots, mobile robots, sensors, and human beings [3]. It is expected that CPS technology will change the approach of human interaction with engineered systems, similarly as the Internet has changed the approach of human interaction with information [4]. Networked and flexible Industry 4.0 environment is a subject of security challenges; therefore, different network participants (human beings as well as technical devices (e.g., sensors), should know if they can trust other nodes in the network. Lately, there has been an increased interest in embedding trust in industry solutions, e.g., using trust-based communication for performance enhancement [5] or efficient resource distribution [6] in industrial IoT.

The research work presented in this paper aims to investigate the concept of trust from several perspectives to derive trust dimensions for Industry 4.0 domain. The addressed research question is "What forms trust models in related research areas and how such models can to be used in Industry 4.0?" To answer this question the concept of trust was analyzed in the social science literature (trust among human beings),

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 43–57, 2018.
https://doi.org/10.1007/978-3-319-97571-9_6

information technology (IT) and computer science literature (how human beings can trust IT solutions), and in ad-hoc distributed sensing systems (trust among physical devices). For each of these areas the trust model was constructed that helped to reveal main trust dimensions in a particular area. The review of related research and derived models can be useful for other researchers and practitioners addressing trust issues in Industry 4.0.

The paper is organized as follows. The related works are discussed in Sect. 2. The proposed models are presented in Sect. 3. Section 4 consists of brief conclusions and points to further research directions.

2 Interpretation of Trust in Industry 4.0 Related Areas

Trust is a vast and widely disputed study topic that originates from the area of social sciences. To gain a multidimensional understanding of the trust concept, its interpretations in social sciences, IT and computer sciences, and ad-hoc distributed sensing systems are investigated in this section.

2.1 Trust in Social Sciences

In social sciences, trust is described by a situation in which a human being (trustor) is willing to rely on chosen actions of another human being (trustee), based on expectation that trustee will behave as desired. A trustor can be harmed if the expectation does not come true [7]. Trust implies readiness to depend on a trustee because of his/her characteristics that express reliability. Rousseau et al. [8] mention three main interconnected trust components: (1) trusting beliefs: confidence in favourable attributes of the trustee that create trusting intentions; (2) trusting intentions: a specific form of a vulnerability state; and (3) trusting behaviours: assured expressions of reliability on and dependency from a trustee.

Nooteboom [9] states that trust takes the form of relational interactions, on a personal level between individuals or on an impersonal level between positions in organisations and institutions. Heimer [10] specifies the fundamental types of trustworthy relations in ascending order: faith, confidence, legal trust, and trust/distrust. Characteristics of trust-based relation types are vulnerability (a trustor possesses valuables that must be entrusted), uncertainty (a trustor realizes that the outcome of interaction with a trustee is uncertain), criticism (a trustee is open for (negative) feedback from a trustor) and alteration: a trustor can influence the trustee or the state of a relationship. Faith as a trust relation type is based purely on the aspect of vulnerability. Interactions that are based on feel of vulnerability and uncertainty, but cannot be addressed to a certain person or organization, are based on confidence. Slightly different trust relationship type is legal trust where the trustee's actions are legally limited and non-negotiable. Trust/distrust is the superior form of a trustworthy relationship. The right to negotiate with a trustee is an added characteristic. Here the trustor feels safe enough to invoke a dispute (trust), or the trustor has secured its assets and feels confident that the trustee will not bring in any harm (distrust).

Hardin separates trust as a phenomenon from concepts of faith, confidence, and regulatory/legal control mechanisms [11]. Mayer et al. add time and emotion dimensions to earlier defined dimensions of individual's ability, benevolence, and integrity. Time dimension refers to time frames in which previous dimensions contribute to trust, as well as merge with or are separate from each other. Perception of antecedents of trust in relationships is influenced by current mood and emotion of a party [7, 12]. Fukuyama in [13] explores the link between trust and cooperation in national economies. Cross-national and cross-cultural differences are likely to affect the degree of reliability on others, especially, initial trust [12].

Trust and reputation are intertwined concepts in social sciences. However, reputation is only sensible in context of community, while trust is applicable also in the case of a single individual [14]. Eisenegger distinguishes three counterparts of reputation – the functional (relevant actions are continuously executed), the social (adherence to social norms is present), and the expressive (comparison with competitors is supported) reputation – that interact to form a common reputation of a trustee [15]. Reputation ties together with the concept of causal influence [9] that is derived from people actions in social relations and contributes largely to trust formation. Continued existence of trustworthy social networks is ensured by encouragement of people who play a role in well-functioning network of relations to express trustworthy behaviour.

2.2 Trust in IT and Computer Science (Human-Computer Interaction)

In IT systems the object of trust is a specific technology or service (abstracted in this section with the concept "computer"). Trust components [8] mentioned in the previous section can be applied to human trust towards IT systems. *Trusting beliefs* of a user are formed from his/her perception of a system in use. *Trusting intention* is a vulnerable state in which the user is before executing a task (e.g., providing a password). Combination of two leads to *trusting behaviour.*

In information systems trust research, IT objects can take different roles – as a mediator between users (a trustor and a trustee) or being a trustee itself [16]. In IT-mediated relationships widely accepted are the following trust formation dimensions from the social science area [7] – *ability, benevolence, and integrity*. They are also used to assess trust. In [17] the authors offer to extend such trust assessment with *predictability* dimension. The IT artefact can also become the focus of trust relationship. More technical dimensions of *performance* (perception of an automated system's capability for supporting user's goals), *process* (user's perception on how appropriate are automated systems algorithms and processes), and *purpose* (user's perception of future value of IT system and evaluation of system designer intentions regarding the user) are introduced to provide insight on trust between a user and an automated IT system in [19]. In [16] formative indicators of these dimensions are introduced: performance dimension – *responsibility*, *information accuracy,* and *reliability*; process dimension – *user authenticity*, *understandability*, *predictability*, *confidentiality,* and *data integrity*; purpose dimension – *authorized data usage, system designer benevolence,* and *faith.*

An example of highly automated human-computer interaction systems are cloud services. Authors of [20] compose trust in cloud services from service consumer perspective; – trust management techniques are based on *policy, recommendation, reputation, and prediction*. Policy technique relies on approaches of credibility and credentials. Credibility approach consists of monitoring and auditing (*fulfilment of SLAs*), entities credibility (measurement of service quantity (e.g. *response time*) and quality (e.g. *availability*, *security*, etc.), and feedback credibility (collected *feedback* from service consumers). Credentials approach controls access levels. Recommendation approach can take a form of either an explicit recommendation (service consumer directly recommends a service to a party that he/she has trusted relations with) or transitive recommendation that occurs when cloud service consumer has a trusted relation with a party that trusts the service, thus extending the personal level of trust towards service provider. Reputation as a trust management technique is built on collected feedbacks from service consumers. Reputation is different from recommendation as service consumers do not have trusted relationships with feedback providers and sources of feedbacks are unknown. Prediction technique can be used when no prior data regarding service interactions exists. It is based on assumption that entities with similar attributes and behaviour have stronger trust relations. Prediction can be used to refine trust results and weight gathered feedback according to relationships that a given entity has with a service provider. In addition, trust characteristics to compare several cloud service providers are defined in [20]: *authentication*, *security*, *privacy, responsibility*, *virtualization,* and *cloud service consumer accessibility.*

Some works distinguish between initial trust (perceptions formed during the first encounter with an unfamiliar automated interface, service provider or e-vendor) and continuous trust (continuous trusting behaviour with services, systems, and interfaces selected by the user). Continuous trust relies on initial trust to determine the extent to which the user will continue his/her trusting behaviours [16]. In [22] authors state that in e-commerce contributing factors to initial trust are *reputation of site vendor, site quality*, and *structural assurance*.

Information system quality aspects, in particular *navigational structure* and *visual appeal*, define the level of trust in IT artefacts, according to [23]. Study field of automated and autonomous systems realizes the importance of user trust in adaptive and self-directed environments. *Benevolence*, *directability, false-alarm rate, perceived competence, reliability, robustness, understandability, utility,* and *validity* are dimensions that form trust in automation setting, according to [24].

2.3 Trust in Ad-Hoc and Distributed Sensing Systems

This section summarizes state-of-art techniques that address trust notion in mobile adhoc networks (MANETs) and wireless sensor networks (WSNs) that are autonomous and distributed systems. Dynamic environments of ad-hoc networks require more complex solutions for trust establishment than traditional cryptographic schemes [25]. One example of trust applications in MANETs and WSNs is trust-based routing. Reputation and recommendation are widely accepted mechanisms to address trust problem in spatially dispersed, scalable mobile networks, in particular, MANETs and WSNs. These metrics are quantitative; most techniques in WSNs and MANETs gather information about node behaviour to evaluate node’s trustworthiness when planning a communication route in the network. As a rule, such models propose to build reputation of each node by enforcing cooperation and gradually excluding maliciously behaving nodes. The resulting mechanism acts as a countermeasure to node misbehaviour and orchestrated denial of service attacks. Two types of trust, direct (also called subjective) trust and indirect trust, are common. Direct trust is local: first-hand neighbour information is obtained. Indirect trust covers broader range of network, i.e. uses trust information from remote nodes. Direct trust is more important than indirect trust for maintaining trust, while indirect trust becomes important for recently added nodes that are not aware of neighbour node behaviours yet [26]. It is possible to combine direct and indirect trust in building total (also referred to as functional) trust [27]. Notions of recommendation and reputation merge in this domain. Nodes share computed trust values, thus forming recommendations; *reputation* of a particular node depends on an amount of positive recommendations. In the area of WSNs and MANETs, *reputation* is often used as a synonym for trust. In early works the notion of trust was not used; instead, reputation-based schemes were proposed.

Reputation-based trust model approaches vary extensively: the use of direct and/or indirect trust values; local or central reputation exchange protocols; exchange of positive or/and negative trust values [26]. For instance, CORE protocol in [27] does not consider negative trust evaluation from indirect nodes to avoid denial of service attacks. CONFIDANT [28] protocol stores trust values in manager modules that oversee the network and broadcast negative values for malicious nodes, thus exposing the network to false reputation attacks. Trust mechanism from [29] also considers historical trust of a node for functional trust value calculation. An overview of techniques for obtaining reputation-based trust is presented in Table 1. Reputation-based trust approaches to node misbehaviour assessment in WSNs are classified as generic, localization, mobility, routing, and aggregation in [30]. Reputation-based framework proposed by [31] is created from two blocks, a Watchdog mechanism that monitors node actions and performs node classification as cooperative and non-cooperative, and a Reputation

Table 1. Approaches to reputation-based trust assessment

Source	Direct trust	Indirect trust	Trust values exchanged: positive (trust)	Trust values exchanged: negative (distrust)	Trust table storage	Trust value range	Node history consideration
[28]	✓	✓	✓	✓	Central	Continuous	
[29]	✓	✓	✓	✓	Local	Continuous	✓
[34]	✓		✓	✓	Local	Continuous	
[31]	✓		✓		Local	Continuous	✓
[27]	✓	✓ (positive values only)	✓	✓	Local	Discrete	
[35]	✓		✓	✓	Local	Discrete	✓

System that maintains the reputation of a node. The Reputation System updates reputation based upon observations coming from Watchdog block, integrates the reputation information based on other available information, adjusts the reputation, and creates an output metric of trust. Rating, weight assignment, probabilistic functions, Bayesian and neural network principles, and Fuzzy logic are some of methods that have been applied by different approaches to trust [36].

Trust value, direct or indirect, is calculated per specific trust function. In [27] baseline functions for investigated MANET are packet forwarding and routing. An approach to assess trust from neighbours' behaviour by calculating packet forwarding ratio in MANETs is researched in [34] where ad-hoc on-demand trusted-path distance vector (AOTDV) protocol is proposed. Similar extended trust-enforcing routing protocol is proposed in [29]. Solution covers *time*, *mobility*, and *successful cooperation frequency* metrics of trust in MANETs.

The model developed in [37] presents a notion of belief and suggests that reliability in ad-hoc network can be ensured without the help of central trust authority. In this work authors derive metrics for trust calculation from information that nodes can passively collect about neighbour elements, such as data packets received/forwarded, control packets received/forwarded, streams established, etc. This information is classified in categories and the trust level in the category can be computed for the node. Total computed trust per node is a sum of weighted categories. Similar metrics are defined in [26]. Trust metric of *reputation* is observed by third parties. A fault tolerant method is proposed in [38] to ensure trust in WSNs. Works in [32, 33] propose alert-based detection mechanisms to identify compromised nodes in sensor networks based on abnormal sensor readings.

In general, reputation-based systems for mobile ad-hoc networks and wireless sensor networks express many common characteristics. The differences between trust models of two network types are in functions (Table 2) used to generate trust ratings.

Table 2. Functions used in reputation-based trust models

Functions	Route discovery	Packet forwarding	Average encounter time	Mobility	Successful cooperation frequency	Sensor readings
Source	[27–29]	[27–29, 34, 35]	[29]	[29]	[29, 35]	[31, 33, 38]

In related works, another large segment of models consider cryptographic methods as trust-establishing mechanisms for secure MANETs and WSNs. In [39] cryptographic approach is separated from trust-based approach in means of ensuring secure MANET by comparing routing protocols. In [40], the authors propose to enhance network security via implementation of trust-based threshold for certificate revocation. Revocation method calculates trust from direct and indirect trust values, using Eigen trust algorithm [41] for direct trust and node's degree of centrality for indirect trust. The authors of [25] suggest using trust to handle soft security threats and to provide secure group communications among uncertain and dynamic nodes in MANETs. In [42], the authors build a more complex trust model for WSNs that consists of several trust dimensions. Every node maintains the trust value about other nodes without central repository for storing trust values of other entities. There are separate trust modules that participate in integrated trust value calculation. Direct trust module derives direct trust value from *communication trust, data consistency* and *energy trust*. Indirect trust module collects recommendations from further nodes to calculate indirect trust value. Integrated trust module establishes a dynamic trust weight function and calculates integrated trust value from previous direct and indirect trust values. Update mechanism module is used to enhance flexibility, i.e. dynamical adaption of the parameters to change the weight sequence to meet the needs of the network. In attempt to make sensor networks resilient to attacks, trust evaluation approach is proposed in [43]. The focus of the work is to evaluate trustworthiness of nodes and filter out deceitful data from compromised nodes. The approach consists of 4 steps: (1) sensor grid is defined based on nodes sensing ranges; (2) claimed location of neighbour nodes is verified using a specific protocol; (3) each sensor entity uses trust evaluation matrix, that contains trust metrics of *identification*, *distance*, *communication ratio*, *sensing result*, *consistency,* and *battery*; and (4) sensing data of multiple nodes are aggregated per grid.

3 Aggregation and Comparison of Trust Factors

In this section, trust beliefs, trust intentions, and trust behaviour factors from social science, IT and computer system, and ad-hoc and distributed sensing system fields are generalized and compared. The results of aggregation and comparison are reflected in conceptual trust models corresponding to the areas of research discussed Sect. 2.

3.1 Trust Model in Social Systems

A multidimensional model of trust that considers various factors from related work discussed in Sect. 2.1 is displayed in Fig. 1. In social systems trusting behaviours are observed between two human participants, or groups of human beings. In the model, *Trustor* object, participant who is willing to show trusting intentions, has distinct properties of vulnerability and uncertainty. Belief factors of Faith and Confidence contribute to Trustor's willingness to trust the other party, either initially or continuously. In addition, internal considerations of Ethical, Cultural and Moral disposition affect trust behaviour of both *Trustor* and *Trustee. Trustee* object, the participant towards which trusting intentions are directed, is described with properties that are perceived by *Trustor*, such as benevolence, emotion, reliability, etc. These are the lower-level metrics that function as second-order formative indicators.

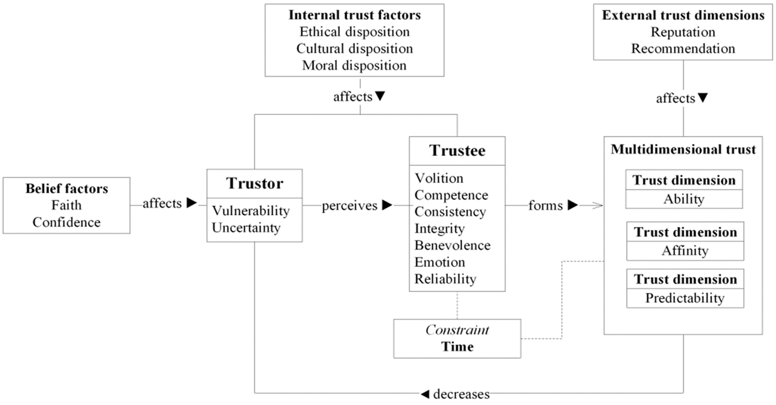


Fig. 1. Multidimensional model of trust in social systems

Higher-level dimensions – ability, affinity, and predictability – are formed from lower-level metrics. These dimensions are not perceived or observed directly by the *Trustor*, i.e. they are composed from second-order formative indicators of *Trustee* properties (volition, competence, etc.). These higher-level trust dimensions form the *Multidimensional trust* object that expresses how willing the *Trustor* is to trust *Trustee*. Both first-order and second-order formative indicators for trust dimensions fall under the effect of, e.g. *time*, which is defined as a constraint in the diagram. Moreover, external trust dimensions of reputation and recommendation affect the *Multidimensional trust* object. *Trustee* does not possess the ability to represent these factors directly or personally; recommendation and reputation are trust dimensions that *Trustor* acknowledges about a specific *Trustee* from a community.

3.2 Trust Model in IT and Computer Science (Human-Computer Interaction)

Related literature review showed that trust in the field of IT and computer science is a complex concept, mostly due to the number of domains in focus and heterogeneity of information systems and their applications. To obtain lowest-level trust metrics for trust criteria comparison, the following steps were executed: (1) lowest-level trust metrics (i.e. stated implicitly as formative factors of trust) discussed in Sect. 2.2 were identified; (2) higher-level trust metrics (i.e. stated as contributors to trust; reflective indicators can be derived) were identified; (3) higher-level metrics were decomposed or extrapolated to lowest-level metrics; (4) common metric pool was created from lowest-level metrics; (5) original and derived lowest-level trust metrics were compared to eliminate duplicates; (6) clarified metric set was classified in trust dimensions of process, performance, and purpose.

Trust metrics are summarized in Tables 3, 4, 5 and 6. Tables 3, 4 and 5 contain observed trust metrics classified by master dimensions. Table 6 contains external trust forming metrics: reputation, recommendation, and feedback related metrics.

Table 3. Trust metrics in performance dimension

Metrics	Responsibility	Accuracy/ validity	Reliability	Fulfillment of SLA	Availability	Response time	Accessibility	Timeliness	Robustness
Source	[16, 20]	[16, 24]	[16, 18, 24]	[20, 21]	[20, 21]	[20]	[20]	[21]	[24]

Table 4. Trust metrics in process dimension

Metrics	Access control	Understandability	Predictability	Confidentiality	Integrity	Privacy	Virtualization	Structure/ directability	Visual appeal
Source	[16, 20]	[16, 24]	[16, 17, 20]	[16]	[16, 17]	[20]	[20]	[22–24]	[23]

Table 5. Trust metrics in purpose dimension

Metrics	Authorized data usage	Designer benevolence	Affinity	Designer initiative	Culture	Competence/ utility/ability
Source	[16]	[16–18, 24]	[21]	[21]	[23]	[17, 24]

Table 6. External trust metrics

Metrics	Reputation	Recommendation	Feedback
Source	[14, 18, 20, 22]	[20]	[20]

Collected and classified metrics were used to represent a trust model in a human-computer interaction system, where *Trustor* object is a human being (system user) and *Trustee* object is an IT artefact. The trust model is illustrated in Fig. 2. In the model *Trustor* object, as in the trust model of social systems in Fig. 1, has distinct properties of vulnerability and uncertainty. Belief factors of Faith, Confidence, Ethnical, Cultural and Moral disposition contribute to Trustor's willingness to trust an IT artefact, either initially or continuously. Higher-level dimensions of *performance* and *process* are perceived from IT artefact's behaviour and can be measured by a system user qualitatively or quantitatively. *Trustee: Performance* object possesses properties that are seen by *Trustor* during system use and can be expressed in terms of statistical values, such as availability, accuracy, reliability, etc. IT artefact's *Trustee: Purpose* object, on the other hand, is a less comprehensive dimension, therefore is apprehended on a perception level by the user. For instance, Designer initiative is based on user's feeling about vendor's or designer's proactivity in solving user problems.

Trustee: Performance and *Trustee: Process* objects are defined as Trust dimension sets in the model. The dimension sets in a given scheme are aggregated elements that contain first-order formative indicators of trust. *Performance* and *process* metrics are collections of actual trust forming dimensions of availability, reliability, understandability, etc. In contrast, social system trust model in Fig. 1 contains lower-level metrics that are second-order formative indicators that lead to first-order formative indicators – actual trust dimensions – ability, affinity, and predictability. *Trustee: Purpose* shares the formation principle with dimensions of ability, affinity, and predictability from social

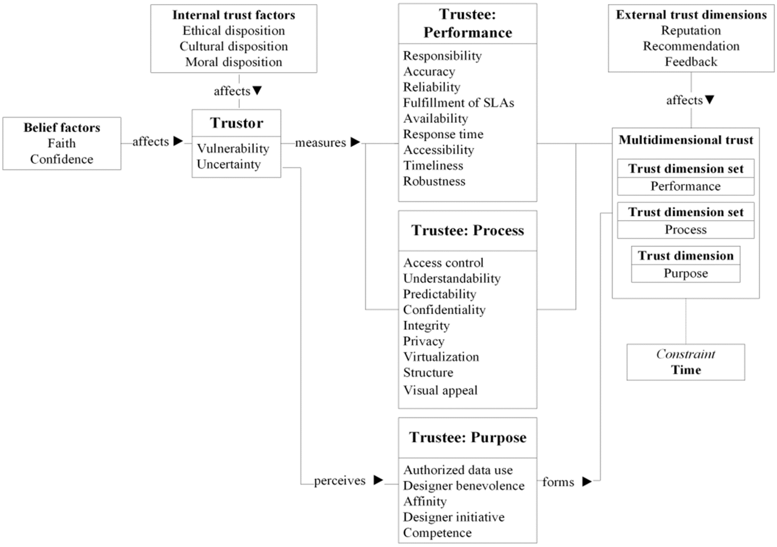


Fig. 2. Multidimensional model of trust in human-computer interaction systems

trust model. It is a trust dimension that contains a first-order formative indicator of *purpose*. In other words, when it comes to interaction with IT artefacts, many more tangible dimensions contribute to trust than if compared with social systems.

Multidimensional trust is a subject to change under influence of, e.g. time, which is defined as a constraint in the diagram. External dimensions of reputation and recommendation are relevant for IT artefacts as well, and are supplemented with the new dimension of feedback.

3.3 Trust Model in Ad-Hoc and Distributed Sensing Systems

A straightforward approach for defining trust factors in CPS industrial wireless network would be to use trust models that have already been designed for needs of mobile ad-hoc networks and wireless sensor networks. However, the literature review in Sect. 2.3 revealed that trust concepts differ in this domain. Very few works, e.g. [42, 43] have made an attempt to develop multidimensional trust assessment models. Problems of reputation and confidentiality have been studied thoroughly by research groups, often indirectly implying that a particular metric is the primary factor for trust as such. Therefore, it can be stated that only certain aspects of trust have been addressed in wireless and ad-hoc mobile networks.

To identify higher level trust dimensions a bottom-up approach was used by grouping trust functions from Table 2 and considering other factors that are not reflected in the table, but are mentioned in related works. Lower-level trust functions were aggregated to define trust dimensions in sensor networks in a setting relevant for this research: (1) lower-level trust metrics were derived from problems that proposed solutions offered to solve them; (2) lower-level metrics were also identified in related works, if stated implicitly; (3) implicit and derived lowest-level trust metrics were compared to eliminate duplicates; (4) unique lower-level metrics were classified in relevant higher-level dimensions; (5) additional human-perspective metrics were determined (the explanation will follow in the remainder of the section).

From lower-level metrics (see, e.g. [26]) the following higher level trust dimensions were derived: dimension of cooperation; dimension of data integrity; and dimension of availability. *Dimension of cooperation* involves monitoring and evaluation of data interactions over the network. This dimension is reached by setting up reputation-based trust models. Aggregated metrics of Route Discovery, Successful Cooperation Frequency, Packet Forwarding, Average Encounter Time, and Mobility from Table 2 contribute to cooperation dimension. Network communication functions that fall into aggregated groups are as follows: Data packets forwarded, Control packets forwarded, Data packet delivery ratio, Control packet delivery ratio, Average end-to-end delay, Throughput, Data streams established, and Timely data transmission. *Dimension of data integrity* is data integrity that considers evaluation of exchanged data consistency. Lower-level trust functions that contribute to data integrity dimension are: Sensor reading consistency, Data packet precision, Control packet precision, and Packet address modification. Regarding *Dimension of availability* certain studies propose to measure the level of trust by evaluating availability, accessibility and similar behavioural properties of particular nodes. Some of availability functions in WSNs and MANETs are: Average time to reach destination node, Power capacity, Responsiveness based on "hello" messages, and Elapsed access time.

A human counterpart also exists in such systems. While not directly participating in information exchange, an individual or a group of people can observe and influence the processes in the network and use the system for certain personal benefits. A human being can also *perceive trust* in any given wireless sensor network. Here, lower-level metrics are observed from human measurements and system observations. Derived lower-level metrics compose the following higher-level dimensions: Dimension of performance; Dimension of process; Dimension of access control; Dimension of communication integrity. *Dimension of performance*: similarly as in human-computer interaction systems from Sect. 2.2, it is possible to evaluate performance of a MANET or WSN by standard monitoring procedures, fault identification mechanisms or service level definitions. Some human-perceived performance trust metrics in WSNs and MANETs are: Availability, Redundancy, Benevolence, Fault tolerance, Ability, Accuracy, and SLA fulfillment. *Dimension of process*: During formal processes of interacting with sensor network, such as node replacement and system configuration, human participant will expect the following factors to be met at a certain level that will contribute to trust towards the system: Predictability, Understandability, Virtualization, Structure, and Configurability. *Dimension of access control*: a human participant can evaluate the security measures established in a wireless network, such as presence of access control mechanisms (authorization, authentication, identification, and accountability), certificate revocation actions, etc. This dimension is defined as a separate module due to autonomous dynamic properties of a network. Lesser control, as perceived by user, forms an extra requirement to stronger security mechanisms. Control of element access to available resources, as well as determination of particular functions, contribute to standard security requirement of data integrity. These functions need to be present in network to satisfy a higher-level *access control* metric. *Dimension of communication integrity*: a human observer can similarly identify if appropriate encryption mechanisms take place in a wireless network. Encrypted messaging ensures basic security needs of data integrity and confidentiality. Encryption and non-repudiation form the *communication integrity* higher-level trust dimension. External trust metrics (reputation as in word-of-mouth) and recommendation are not considered in case of wireless sensor networks, since other human recommendations about certain sensor network are not relevant in such setting.

Collected and classified trust dimensions and lower-level metrics were used to represent trust model of ad-hoc and distributed sensing systems in Fig. 3. In this model there are two *Trustor* objects: one is human being (sensor system observer or maintenance responsible); another object is the node that is a participant of a network. *Trustee* object is the sensor network as a whole, an integral entity. Human *Trustor* object shares behavioural properties similar to ones seen in Fig. 2, in human-computer interaction scheme. For *Trustor* node (on the left in Fig. 3) the property of vulnerability is left and uncertainty is removed. Arguably, uncertainty is an emotion applicable to human beings only. We assume that the node is not capable of expressing uncertain at initiation moment. Over time, when trust mechanisms are implemented, uncertainty property for a node regarding another node can arise by, e.g. enabling reputation tables in the system. Each node, however, is vulnerable to attacks at any given moment. Lower-level trust metrics are grouped in three trust objects: *Trustee: Cooperation, Trustee: Data integrity, Trustee: Availability*. These trust dimensions can only be computed by monitoring neighbouring node behaviour.

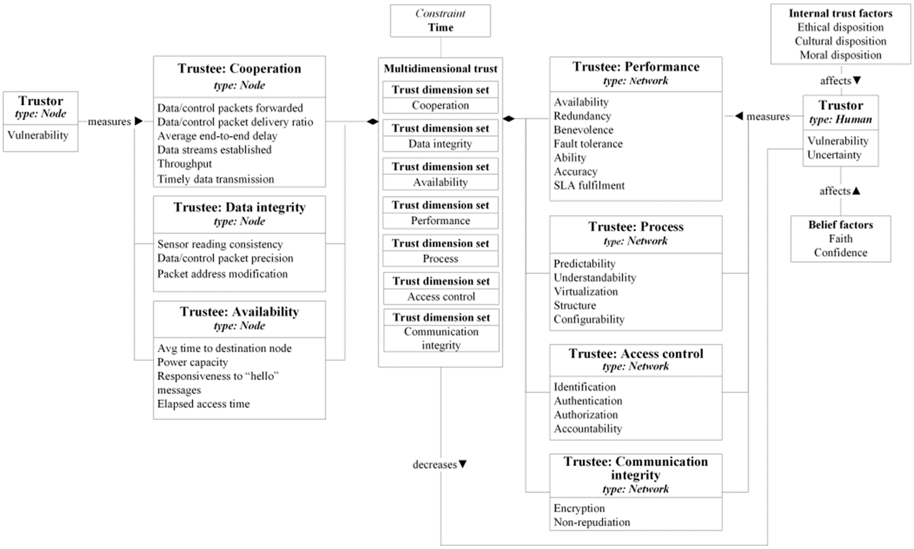


Fig. 3. Multidimensional model of ad-hoc and distributed sensing systems

All seven trust dimension sets: *Cooperation*, *Data integrity, Availability, Performance, Process, Access control*, and *Communication integrity* are mandatory constituents for *Multidimensional trust* module. Dimension sets are aggregated modules that contain first-order formative indicators of trust. *Performance* or *Cooperation* metric does not contribute to trust per se; instead, lower-level functions such as sensor reading consistency and data packet delivery ratio are used to establish the trust in the network. If compared with trust model for human-computer interaction systems, here all described metrics are *quantitative*.

Lastly, *Multidimensional trust* object falls under the influence of time, which is denoted as a constraint in the diagram. Dimensions of reputation and recommendation are not considered in the model as separate entities. Higher-level dimensions of cooperation and data integrity represent the node reputation already. Recommendation can also be expressed by indirect trust evaluation from remote nodes.

4 Conclusions

The paper provides preliminary results of the research aimed at identifying trust dimensions for Industry 4.0. The interpretations of trust in three areas (social science, IT and computer science, and ad-hoc and distributed sensing systems) were amalgamated and analyzed. On the basis of the analysis, the trust models for each area were developed and the main trust dimensions were revealed. Namely, according to the preliminary results, the main trust dimensions for situations were trust is established among human beings are Ability, Affinity, and Predictability; for situations where a human being has to trust an IT solution, the main dimensions are Performance, Process, and Purpose. For establishing trust in ad-hoc and distributed sensing systems, the following main trust dimensions are to be taken into account: Cooperation, Data integrity, Availability, Performance, Process, Access control, and Communication integrity. It is a matter of further research to organize the proposed trust models and trust dimensions in an integrated trust model for Industry 4.0, as well as to perform evaluation of its forming components, presented in this paper. We acknowledge that this research was partly done within the Erasmus+ Strategic Partnership “Improving Employability through Internationalisation and Collaboration” (EPIC) project, with the support of the Erasmus + programme of the European Union.

References

1. Kagermann, H., Wahlster, W., Helbig, J.: Securing the future of German manufacturing industry. Acatech (2013)
2. i-SCOOP: 17 March 2018. https://www.i-scoop.eu/industry-4-0/
3. Li, X., Li, D., Wan, J., Vasilakos, A.V., Lai, C.F., Wang, S.: A review of industrial wireless networks in the context of Industry 4.0. Wirel. Netw. **23**(1), 23–41 (2017)
4. National Science Foundation, Cyber-Physical Systems (CPS): 27 April 2018. https://www.nsf.gov/pubs/2018/nsf18538/nsf18538.pdf
5. Zhu, C., Rodrigues, J.J.P.C., Leung, V.C.M., Shu, L., Yang, L.T.: Trust-based communication for the industrial Internet of Things. IEEE Commun. Mag. **56**(2), 16–22 (2018)

6. Jeong, S., Na, W., Kim, J., Cho, S.: Internet of Things for smart manufacturing system: trust issues in resource allocation. IEEE Internet Things J. (Early Access). i-SCOOP, 17 March 2018. https://www.i-scoop.eu/industry-4-0/
7. Mayer, R.C., Davis, J.H., Schoorman, F.D.: An integrative model of organizational trust. Acad. Manag. Rev. **20**(3), 709–734 (1995)
8. Rousseau, D.M., Sitkin, S.B., Burt, R.S., Camerer, C.: Not so different after all: a cross-discipline view of trust. Acad. Manag. Rev. **23**(3), 393–404 (1998)
9. Nooteboom, B.: Trust: Forms, Foundations, Functions, Failures and Figures. Edward Elgar, Cheltenham (2002)
10. Heimer, C.A.: Solving the problem of trust. In: Cook, K.S. (ed.) Trust in Society, pp. 40–88. Russell Sage Foundation, New York (2001)
11. Hardin, R.: Conceptions and explanations of trust. In: Cook, K.S. (ed.) Trust in Society, pp. 3–39. Russell Sage Foundation, New York (2001)
12. Schoorman, F.D., Mayer, R.C., Davis, J.H.: An integrative model of organizational trust: past, present, and future. Acad. Manag. Rev. **32**(2), 344–354 (2007)
13. Fukuyama, F.: Trust: The Social Virtues and the Creation of Prosperity. Free Press, New York (1995)
14. Trček, D.: A brief overview of trust and reputation over various domains. Trust and Reputation Management Systems. SIS, pp. 5–19. Springer, Cham (2018). https://doi.org/10.1007/978-3-319-62374-0_2
15. Eisenegger, M.: Trust and reputation in the age of globalisation. In: Klewes, J., Wreschniok, R. (eds.) Reputation Capital. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-01630-1_2
16. Söllner, M., Hoffmann, A., Hoffmann, H., Wacker, A., Leimeister, J.M.: Understanding the formation of trust in IT artifacts. In: Proceedings of the International Conference on Information Systems, Orlando, Florida (2012)
17. Gefen, D., Karahanna, E., Staub, D.W.: Trust and tam in online shopping: an integrated model. MIS Q. **27**(1), 51–90 (2003)
18. Talboom, S., Pierson, J.: Understanding trust within online discussion boards: trust formation in the absence of reputation systems. In: Fernández-Gago, C., Martinelli, F., Pearson, S., Agudo, I. (eds.) IFIPTM 2013. IAICT, vol. 401, pp. 83–99. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38323-6_6
19. Lee, J., Moray, N.: Trust, control strategies and allocation of function in human-machine. Ergonomics **35**(10), 1243–1270 (1992)
20. Noor, T.H., Sheng, Q.Z., Bouguettaya, A.: Trust Management in Cloud Services. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-12250-2
21. Pawar, P.S., Rajarajan, M., Dimitrakos, T., Zisman, A.: Trust model for cloud based on cloud characteristics. In: Fernández-Gago, C., Martinelli, F., Pearson, S., Agudo, I. (eds.) IFIPTM 2013. IAICT, vol. 401, pp. 239–246. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-38323-6_18
22. McKnight, D.H., Choudhury, V., Kacmarc, C.: The impact of initial consumer trust on intentions to transact with a web site: a trust building model. J. Strateg. Inf. Syst. **11**, 297–323 (2002)
23. Vance, A., Elie-Dit-Cosaque, C., Straub, D.: Examining trust in information technology artifacts: the effects of system quality and culture. J. Manag. Inf. Syst. **24**(4), 73–100 (2008)
24. Palmer, G., Selwyn, A., Zwillinger, D.: The “Trust V”: building and measuring trust in autonomous systems. In: Mittu, R., Sofge, D., Wagner, A., Lawless, W.F. (eds.) Robust Intelligence and Trust in Autonomous Systems, pp. 55–77. Springer, Boston, MA (2016). https://doi.org/10.1007/978-1-4899-7668-0_4

25. Janani, V.S., Manikandan, M.S.K.: Efficient trust management with Bayesian-evidence theorem to secure public key infrastructure-based mobile ad hoc networks. EURASIP J. Wirel. Commun. Netw. **2018**, 25 (2018)
26. Zahariadis, T., Leligou, H.C., Trakadas, P., Voliotis, S.: Trust management in wireless sensor networks. Trans. Emerg. Telecommun. Technol. **21**(4), 386–395 (2010)
27. Michiardi, P., Molva, R.: Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. In: Jerman-Blažič, B., Klobučar, T. (eds.) Advanced Communications and Multimedia Security. ITIFIP, vol. 100, pp. 107–121. Springer, Boston, MA (2002). https://doi.org/10.1007/978-0-387-35612-9_9
28. Buchegger, S., Boudec, J.Y.L.: Performance analysis of the CONFIDANT protocol (Cooperation of Nodes – Fairness in Dynamic Ad-hoc NeTworks). In: The 3rd ACM International Symposium Mobile Ad-hoc Networking & Computing (MobiHoc 2002), Lausanne, CH (2002)
29. Feng, R., Che, S., Wang, X.: A credible routing based on a novel trust mechanism in ad hoc networks. Int. J. Distrib. Sens. Netw. **9**(4), 652051 (2013)
30. Li, X., Jia, Z., Zhang, P., Zhang, R., Wang, H.: Trust-based on-demand multipath routing. IET Inf. Secur. **4**(4), 212–232 (2010)
31. Ganeriwal, S., Balzano, L., Srivastava, M.: Reputation-based framework for high integrity sensor networks. ACM Trans. Sens. Netw. **4**, 1–37 (2008)
32. Zhaoyu, L., Joy, A.W., Thompson, R.A.: A dynamic trust model for mobile ad hoc networks. In: IEEE International Workshop on Future Trends of Distributed Computing Systems (2004)
33. Zhang, Q., Yu, V., Ning, P.: A Framework for Identifying Compromised Nodes in Sensor Networks. In: Securecomm and Workshops (2006)
34. Yao, Z., Kim, D., Doh, Y.: PLUS: Parameterized and localized trUst management scheme for sensor networks security. In: International Conference on Mobile Ad Hoc and Sensor Systems (2006)
35. Momani, M., Challa, S.: Survey of trust models in different network domains. Int. J. Ad Hoc Sens. Ubiquit. Comput. **1**(3), 1 (2010)
36. Pirzada, A.A., McDonald, C.: Establishing trust in pure ad-hoc networks. In: The 27th Australasian Conference on Computer Science, Dunedin, New Zealand (2004)
37. Krasniewski, M., Varadharajan, P., Rabeler, B., Bagchi, S., Hu, Y.C.: TIBFIT: Trust index based fault tolerance for arbitrary data faults in sensor. In: International Conference on Dependable Systems and Networks (DSN 2005), Yokohama, Japan (2005)
38. Cordasco, J., Wetzel, S.: Cryptographic versus trust-based methods for MANET routing security. Electron. Notes Theor. Comput. Sci. **197**(2), 131–140 (2008)
39. Rajkumar, B., Narsimha, D.G.: Trust based certificate revocation for secure routing. Procedia Comput. Sci. **92**, 431–441 (2016)
40. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The Eigentrust algorithm for reputation management in P2P networks. In: Proceedings of the 12th international conference on World Wide Web, WWW 2003, Budapest, Hungary (2003)
41. Ye, Z., Wen, T., Liu, Z., Song, X., Fu, C.: An efficient dynamic trust evaluation model for wireless sensor networks. J. Sens. **2017**, 16 (2017)
42. Hur, J., Lee, Y., Yoon, H., Choi, D., Jin, S.: Trust evaluation model for wireless sensor networks. In: The 7th International Conference on Advanced Communication Technology, Phoenix Park, South Korea (2005)
43. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. Decis. Support Syst. **43**(2), 618–644 (2007)

Towards the Reference Model for Security Risk Management in Internet of Things

Raman Shapaval and Raimundas Matulevičius(✉)

Institute of Computer Science, University of Tartu, Tartu, Estonia
shapovalromeo@gmail.com, raimundas.matulevicius@ut.ee

Abstract. Security in the Internet of Things (IoT) systems is an important topic. In this paper we propose an initial comprehensive reference model to management security risks to the information and data assets managed and controlled in the IoT systems. Based on the domain model for the information systems security risk management, we explore how the vulnerabilities and their countermeasures defined in the open Web application security project could be considered in the IoT context. To illustrate applicability of the reference model we analyse how reported IoT security risks could be considered.

Keywords: Internet of Things (IoT)
Information Systems Security Risk Management (ISSRM)
Open Web Application Security Project (OWASP)

1 Introduction

Internet of Things (IoT) is a network of connected devices and systems to exchange or accumulate data and information generated by users of and embedded sensors in the physical objects [8]. Among the privacy, energy-awareness, environment, and other concerns, security plays an important role, as the (potentially sensitive) data is sent among the various devices and multiple users. In cases where such a data is intercepted and used for non-intended purposes, it may lead to the severe damages of the valuable system and/or environmental assets [7,10,12,15–17]. There exist a number of surveys related to the IoT security [1,2], security of the IoT frameworks [3,18], or specific components of the IoT systems [4,9,11]. In this paper we propose a comprehensive reference model for the *security risk management in the IoT systems*. We base our proposal on the domain model for the information systems security risk management (ISSRM) [6,13] – thus, we focus on the security risks to the information and data managed in the IoT system. Since the IoT systems much depend on the cloud and Internet computing we consider how vulnerabilities and their countermeasure considered in the open Web application security project (OWASP) [14] can help when identifying and managing the security risks in the IoT systems.

The rest of the paper is structured as follows: in Sect. 2 we overview the ISSRM domain model. Section 3 presents components for managing IoT security

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 58–72, 2018.
https://doi.org/10.1007/978-3-319-97571-9_7

risks. This includes discussion on the IoT assets, their vulnerabilities and countermeasures used to mitigate these vulnerabilities. Section 4 gives few examples illustrating some reported security risks. Finally, Sect. 5 concludes the paper and provides directions for future work.

2 Domain Model for Security Risk Management

The ISSRM domain model (see Fig. 1) suggests three conceptual pillars to explain secure *assets*, *security risks* and their *countermeasures* [6,13]. Here, the *business asset* is understood in terms of the information, data and processes, which bring value to the organisation. Business assets are supported by the *system assets* (a.k.a., IS assets). *Security criteria* (i.e., confidentiality, availability, and integrity) are the constraints of the business assets and define the security needs. *Security risk* is defined as a combination of the *event* and *impact*. Here, *impact* negates the security criterion and harms at least two (one system and one business) assets. Event is defined in terms of *threat* and *vulnerability*. A *vulnerability* is a characteristic of the system assets and it constitutes a weakness of this asset. A *threat* targets the systems assets by exploiting its vulnerability. Threat is defines as combination of the *threat agent*, an active entity who has interest to harm the assets, and the *attack method*, the means used to carry on the threat. Security risk treatment concepts include risk treatment decision, security requirements, and controls. *Security risk treatment* is a decision to treat the identified risk. It is refined to the *security requirements*, which define the condition to be reached by mitigating the security risks. Finally the *controls* implement the defined security requirements.

In this paper we will use the ISSRM domain model to combine constituencies of the IoT system security risks.

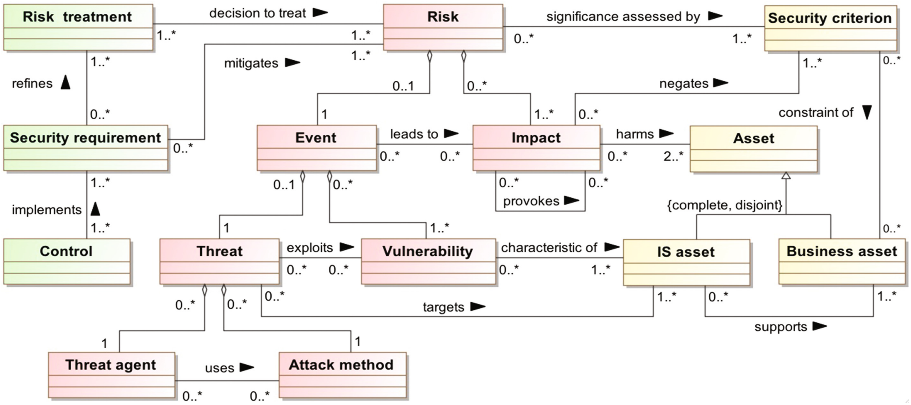


Fig. 1. The ISSRM domain model, adapted from [6,13]

3 Security Risk Management in IoT Systems

3.1 Content and Assets

Let's represent some process in a simple IoT system, illustrated in Fig. 2. Here, four IoT sensors are illustrated (*i*) temperature sensor - detects heat; (*ii*) Wi-Fi module - sends detected temperature from the sensor to the control center; (*iii*) control center - makes decision what command should be sent to the sprinkler based on the previously received temperature from the temperature sensor; (*iv*) actuator - represented by sprinkler, turns on and puts out flame or turns off depending on the command from the control center. Let's suppose that an attacker with means to break a normal work flow of this IoT system exploits the weakness of the Wi-Fi connection between Wi-Fi module and Control center. This way he gets an access to the temperature which is going from the sensor. Then an attacker changes the temperature value and this event leads to the overheating and consequently it harms the sprinkler.

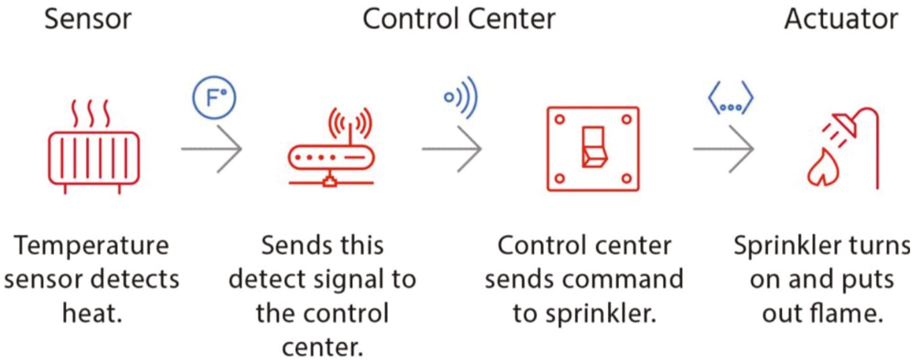

Sensor
Control Center
Actuator
F°
Temperature sensor detects heat.
Sends this detect signal to the control center.
Control center sends command to sprinkler.
Sprinkler turns on and puts out flame.

Fig. 2. Example of IoT sensors

Figure 3 presents an IoT architecture model [5]. Here, the *IoT system* consists of *service* used by the *user*, remote or/and local *storage*, and *computing device*. There exists an *IoT device*, which interacts with the *computing device*. Different *computing devices* are connected to each other. *IoT devices* manage some *entities*, which can be either on-device and/or network *resources*. *Remote storage* and *network recourses* are placed on the *cloud* environment.

The IoT architecture provides the IoT components which correspond to the system and business assets. The IoT assets is anything that is valuable for the IoT system or play an important role in providing functionality and services to users. Like in [6,13] the IoT system assets gain their importance in supporting the business assets. Thus, they can be represented as ground components of the information technology such as hardware, software or network. For example in Fig. 2 system assets are sensors, Wi-Fi, control center and actuator.

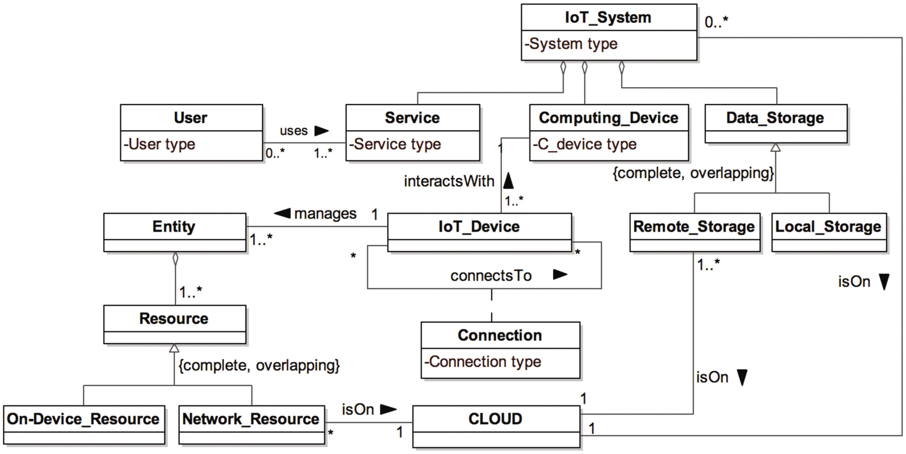


Fig. 3. IoT architecture model

Business assets are valuable for each IoT system as they represent essential business value such as information, processes, capabilities and skills [6,13]. Besides official definitions, business assets can be commonly represented by the data, which is transferred, stored or manipulated in the IoT system during working process. As a result business assets security is defined in terms of security criteria (i.e., confidentiality, integrity or availability). For instance in Fig. 2 business assets are temperature which goes from the Sensor to the Control center, and commands given by the Control center to the Actuator.

3.2 IoT Vulnerabilities

In this section we present ten system vulnerabilities which can be observed in the IoT systems. Typically, vulnerability is presented as a weakness in a design flaw or an implementation bug. They allow an attacker to harm applications, users, and other entities that rely on this application. As the IoT systems are using the Web applications, the vulnerabilities concerned in the OWASP project [14] could be seen as the potential ones in the IoT systems.

For each vulnerability type, we present its definition, its discovery options, and the major threat agent's profile to exploit the vulnerability. In majority of the cases if successful, the risk event would lead to negation of service's data confidentiality, integrity and availability, harm or taking over the IoT device and compromise of the IoT System along with the users and their data.

Insecure Web Interface, V#1: The most common vulnerabilities of the insecure Web Interface are account enumeration, lack of account lockout and weak credentials. They can be discovered during manual system testing or by using testing tools for cross-site scripting identification. The vulnerability can be exploited by an attacker (internal or external) with an access to the web interface, who uses weak credentials, gains access to plain-text credentials or enumerates accounts to access the web interface.

Insufficient Authentication and/or Authorisation, V#2: These vulnerabilities include weak passwords or credentials with poor protection which could lead to insufficient user authentication and/or authorisation. The insufficient authentication and/or authorisation vulnerabilities can be discovered while examining the interface of the system with automated testing and require more stable passwords. An attacker (e.g., internal or external user) with the access to the web interface uses weak passwords, insecure password recovery mechanisms, poorly protected credentials or lack of granular access control to access a particular interfaces of the IoT devices.

Insecure Network Services, V#3: These vulnerabilities are not sufficient controls of open ports and weak traffic monitoring in the IoT system. The insecure network service vulnerabilities are discovered using port scanner tools and fuzzers[1]. Here, an attacker (internal or external IoT user) with a network access to the IoT device exploits these vulnerabilities in network services to launch an attack on IoT device itself or bounce attacks off the device.

Lack of Communication Encryption, V#4: This vulnerability is a rather common in local networks as it is assumed that the network traffic will not be widely visible and/or accessible. However, in such a network, the data transfer is visible in a range of the wireless network support or using/having an external access to the local network. Communication encryption vulnerabilities are rather easy to discover simply by launching a testing attack on the IoT system with viewing network traffic and searching for readable data in it. An automated tools can also look for proper implementation of common communication encryption such as SSL and TLS. An attacker with the access to the network the IoT device uses the lack of communication encryption to view data being passed over the network.

Privacy Concerns (Confidentiality), V#5: These vulnerabilities comprises authentication/authorization, weakly protected communication protocols, inappropriate secure network services. Privacy concerns are relatively easy to discover by reviewing the collected data of when IoT device is being setup and activated by the user. One can use automated tools to search for specific data patterns that indicate personal/sensitive data. An attacker (with the access to the IoT device, network the device is connected to, the mobile application, and cloud) uses insufficient authentication, lack of transport encryption or insecure network services to negate IoT system data confidentiality.

Insecure Cloud Interface, V#6: This vulnerability occurs because of the low security level of the cloud access credentials or account enumeration. Insecure

[1] "Fuzzing or fuzz testing is an automated software testing technique that involves providing invalid, unexpected, or random data as inputs to a computer program. The program is then monitored for exceptions such as crashes, or failing built-in code assertions or for finding potential memory leaks. Typically, fuzzers are used to test programs that take structured inputs." https://en.wikipedia.org/wiki/Fuzzing.

cloud interfaces are discovered by reviewing the connection to the cloud interface and by identifying if SSL is in use. Another way is to use password reset mechanism to identify valid accounts which can lead to account enumeration. An attacker with an access to the internet uses insufficient authentication, lack of transport encryption and account enumeration to access data or take over the controls via the cloud interface (i.e., Website).

Insecure Mobile Interface, V#7: This vulnerability occurs because of the low security level of the app access credentials or account enumeration. Insecure mobile interfaces are discover by reviewing connection to the wireless networks and identifying if SSL is in use. Another way is to use the password reset mechanism to identify valid accounts which can lead to account enumeration. An attacker with the access to the mobile application, uses insufficient authentication, lack of transport encryption and account enumeration to access data or to take over the IoT system controls via the mobile interface (application).

Insufficient Security Configurability, V#8: These vulnerabilities appears when users of the IoT device have limited or no ability to alter its security controls; for example, when the web interface does not support creating granular user permissions and does not force user for creating strong password. These insufficient security configuration vulnerabilities are discover by reviewing the web interfaces along with the provided options. An attacker with an access to the IoT device, exploits a lack of granular permissions to access data or controls on the device, low level transport encryption. The use of the weak passwords could also be the doors for the attack on the IoT system.

Insecure Software and/or Firmware, V#9: These vulnerabilities relate to the insufficient IoT system states' monitoring and weak encryption. The IoT devices should be updated with the new patches of the software and/or firmware. However the update patches can be harmed while transferring them using the unprotected channels. In addition, both software and firmware can also be insecure if they hardcode sensitive/personal data. Security issues with software and/or firmware are discovered by inspecting the network traffic during the update, by checking encryption, or by using a hex editor to inspect the update files regarding the targeted information.

An attacker (with an access to the IoT device, network the device connects to, and/or the server, which provides updates to the IoT devices software) captures files with updates from unencrypted connection between the server and IoT devices. The files with updates are not encrypted. Thus the attacker is able to perform unauthorised update of the IoT system software via DNS hijacking.

Poor Physical Security, V#10: These weaknesses are present when an attacker is able disassemble a device to easily access the storage medium and any data stored on that medium. Poor physical security vulnerabilities are also present when USB ports or other external ports can be used to access the device using features intended for configuration or maintenance. These vulnerabilities are discovered by checking hardware components. An attacker who has physical access to any physical IoT system asset, uses USB ports, SD cards or other storage means to access the operating system and potentially any data stored on the IoT device.

3.3 IoT Security Countermeasures

In this section we identify a set of countermeasures to mitigate the security risks to the IoT systems. It is important to note that selection of the concrete countermeasures much depends on the security risk treatment decision made (e.g., risk reduction, risk avoidance, risk transfer and risk retention), trade-off analysis, and return on security investment analysis results. Below we group security countermeasures to five groups: (*i*) protocol and network security; (*ii*) data and privacy; (*iii*) identity management; (*iv*) trust and governance; and (*v*) fault tolerance.

Protocol and Network Security. Since cryptography plays leading role in developing security protocols for securing network infrastructure, protocol and network security often requires optimisations in cryptography algorithms and key management systems. However, applying standard Internet security mechanisms to the IoT system assets could be inefficient due to the lack of the assets' resources. Therefore, security protocols should be adapted based on the actual IoT system architecture taking into account their performance and capabilities. According to the OWASP [14], the following security countermeasure could be considered for the protocol and network security:

1. ***Secure network services***, **Cm#1:**
   - Only the necessary ports which are important for the IoT system functionality, should be exposed and available;
   - Service should be protected from buffer overflow (overrun) and fuzzing attacks [14];
   - One should ensure that the service is prepared for and not vulnerable against the DoS attacks; at least some monitoring system should be established to determine such an attack type;
   - One should ensure that network ports and services are exposed to the internet via UPnP;
   - Monitoring system should block abnormal service requests.

2 ***Communication encryption***, **Cm#2:**
   - SSL and TLS (or other similar) communication protocols should be used to transfer encrypted data through the network;
   - One should use the encryption techniques;
   - The MQTT payload encryption should be used to protect IoT system's data on the application level;
   - One should ensure the secure encryption key handshaking.

Data and Privacy. Data and privacy keys support the sensitive and valuable assets in the IoT system. The users personal data should be protected. However there exists systems that can be used to target management of the users' data. This means one need to monitor the data managers assigned for the personal data. Consequently, it is necessary to establish cryptographic algorithms and protocols for securing data transfers. In other words there should exists data management policies regarding different types of the personal data. Following [14], the following countermeasures could be set for data privacy management:

1. ***Privacy concerns***, **Cm#3:**
   - One should ensure that (*i*) only data critical to the functionality of the IoT system is collected; (*ii*) sensitive data is not collected; (*iiii*) collected data is de-identified and/or anonymised; (*iv*) retention limits are set for the collected data; (*v*) collected data is encrypted.
   - One should ensure that personal information is properly protected at the different components of the IoT system;
   - Only authorised users should have access to the collected data;
   - Only data needed to audit certain functionality of the IoT system should be collected;
   - One should de-identified collected data before analysing.
2. ***Secure software and/or firmware***, **Cm#4:**
   - One should ensure that the system uses secure update mechanism and all files transferring is based on accepted encryption methods;
   - The update file should not expose sensitive data;
   - Each pack of incoming files with updates should be signed and verified before saving it in the IoT system's memory storage.
   - One should use only trusted and secure servers for updates.
   - One should use secure boot[2].
3. ***Physical security***, **Cm#5:**
   - One should use only trusted and protected data storage services;
   - Stored data should be encrypted;
   - One should protect the USB (or other external) ports (from uploading malicious software);
   - One should minimise number of external ports (e.g., USB) used in the IoT system;
   - One should ensure that the IoT system has "the ability to limit administrative capabilities" [14].

Identity Management. Identity management represents a non-trivial process of verifying a staggering variety of identity and connection types. The following rules should be followed:

1. An object's identity should always be unique compared to the other objects from its family.
2. Unique identity should be called core identity, as an object can also have several temporary identities.
3. Self-identification is one of the ground features of an object.
4. An object should know the identity of its owner.

[2] "Secure Boot is a technology where the system firmware checks that the system boot loader is signed with a cryptographic key authorised by a database contained in the firmware" https://docs-old.fedoraproject.org/.

An important mechanism is to give an object opportunity to cover its identity if needed. As the IoT systems cover different user's profiles, there are situations when it is not secure to reveal identities only based on incoming requests. The following countermeasures could be set for the identify management [14]:

1. ***Secure authentication and/or authorisation*, Cm#6:**
   - The IoT system should insists for the "strong" usernames and passwords;
   - Two factor authentication and users' credentials encryption should be implemented;
   - Insecure password recovery mechanisms should be restricted; re-authentication should be "required for sensitive features" [14].
   - Revoking mechanism should be developed for the IoT system's credentials;
   - Application, device and server authentication should be required;
   - "Manage authenticated user id (i.e., credential info) and the user's device id, the user's app id mapping table in the authentication server" [14].
   - Authentication token and/or sessions should always be unique to each user along with user id, app id and device id.
2. ***Secure Web interface*, Cm#7:**
   - Default username and password should be non trivial and its recommended to change them during initial setup.
   - "Forgot password" functionality should be secure and sturdy. One should not provide user the information indicating a valid account;
   - One should check if the Web interface is protected against the XSS (Cross-site Scripting), SQLi (SQL injections) and CSRF (Cross-site request forgery) attacks;
   - One should use encrypted communication protocols while transferring IoT system's credentials;
   - One should restrict usage of "weak" passwords;
   - One should set a predefined number of attempts to log to the IoT system.

     If the number is exceeded one should (temporary) block the user for further authentication procedures.
3. ***Secure mobile interface*, Cm#8:**
   - Default usernames and passwords should be changed during initial setup;
   - One should ensure that the IoT system is protected from account enumeration through password reset mechanisms;
   - One should set a limit for the unsuccessful login attempts. If it is reached, the user should be (temporary) blocked;
   - One should not transfer the IoT system's credentials over the Internet;
   - Two factor authentication should be implemented;
   - One should use obfuscation techniques applied to mobile app;
   - One should restrict the mobile app's execution on tempered OS environment [14].

Trust and Governance, Cm#9: Trust and governance form the fundamentals in the IoT system. Represented by mechanism, which dynamically evaluates objects, it helps to control users' services based on the interaction process. In pair with governance, trust supports cohesion and stability of the security protocols in the IoT system which much relies on the Cloud-based services. According to [14] we can recommend next countermeasures:

- Default usernames and passwords should always be changed during initial setup;
- The IoT system should be protected from account enumeration through password reset mechanisms;
- One should set a limit for the unsuccessful login attempts (so that the user's is (temporary) blocked);
- Cloud-based interface should be always protected from XSS (Cross-site Scripting), SQLi (SQL injections) or CSRF (Cross-site request forgery) attacks;
- The IoT system's (user's) credentials should not be transferred over the Internet;
- The two factor authentication should be implemented;
- Monitoring system should block abnormal service requests.

Fault Tolerance, Cm#10: The IoT system can't be entirely secured, since the number of possible security risks is increasing faster then the number of solutions for covering these newly appearing risks. Accomplishing acceptable fault tolerance in IoT requires next interdependent efforts:

- Objects have to be secure "by default", it means that beside secure protocols and algorithms the software structure should be improved.
- The state of the network and its services should be shared among the IoT objects. This will provide an opportunity to maintain states' monitoring management on the needed level as these objects would be able to associate their own state changes with the network state changes. Constituently, the system monitoring quality will grow, too.
- Linked to the second effort, the object should be able to protect itself if the network state informs about the network collapse or security threat. To arrange this functionality, the intrusion-detection system (or other defensive tools) should be introduced.
- Functionality of the users and administrators should be separated;
- The IoT system should be able to encrypt data at rest or in transit;
- Users must use "only strong passwords" when authorising to the IoT system;
- The system should log the security events and notify end users about them.

3.4 Reference Model of IoT Security Risk Management

In Fig. 4 we combine IoT system assets discussed in Sect. 3.1, IoT vulnerabilities overviewed in Sect. 3.2 and their countermeasure presented in Sect. 3.3 to a comprehensive reference model for the IoT security risk management. Firstly,

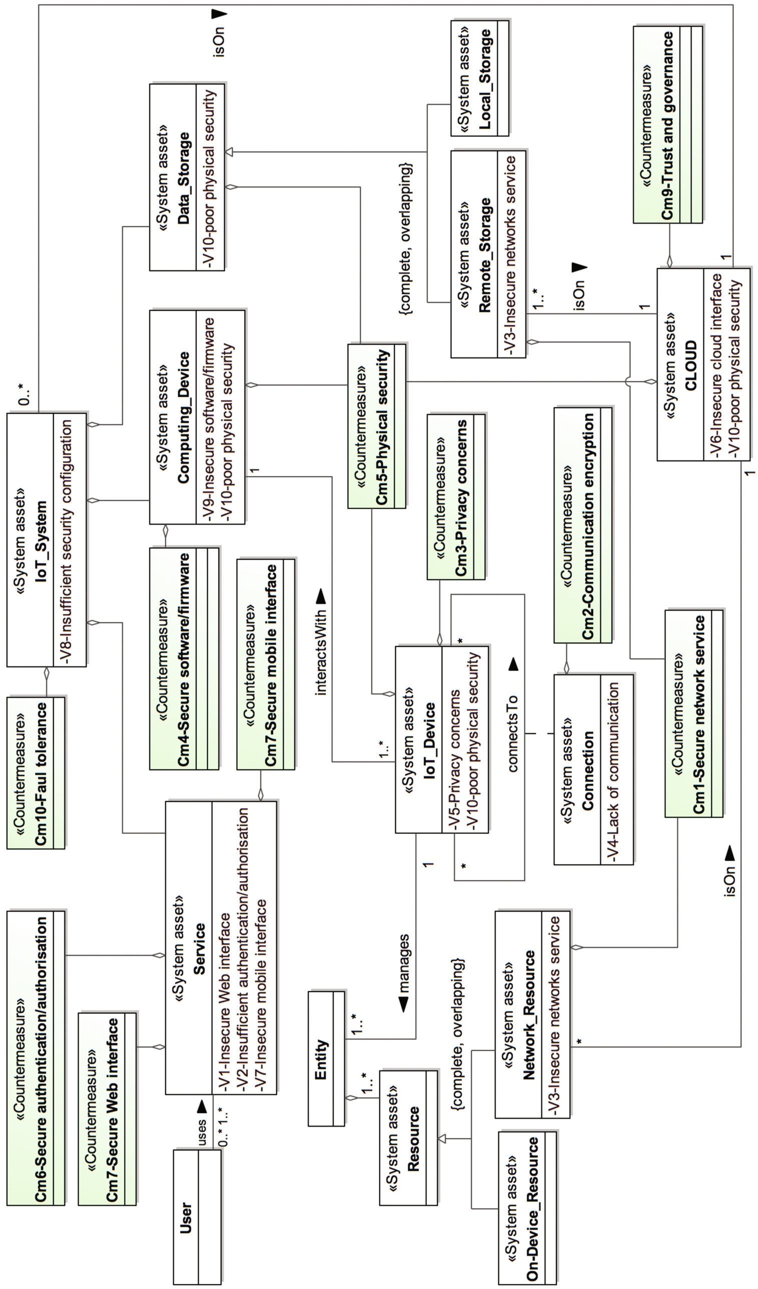

«System asset» IoT_System
-V8-Insufficient security configuration
«Countermeasure» Cm10-Faul tolerance
«System asset» Data_Storage
-V10-poor physical security
«System asset» Computing_Device
-V9-Insecure software/firmware
-V10-poor physical security
«Countermeasure» Cm4-Secure software/firmware
«Countermeasure» Cm7-Secure mobile interface
«System asset» Service
-V1-Insecure Web interface
-V2-Insufficient authentication/authorisation
-V7-Insecure mobile interface
«Countermeasure» Cm6-Secure authentication/authorisation
«Countermeasure» Cm7-Secure Web interface
User
uses
0..*
1..*
Entity
1..*
1..*
manages
«System asset» Resource
{complete, overlapping}
«System asset» Network_Resource
-V3-Insecure networks service
«System asset» On-Device_Resource
«Countermeasure» Cm5-Physical security
interactsWith
1
1..*
«System asset» IoT_Device
-V5-Privacy concerns
-V10-poor physical security
1
*
*
connectsTo
«Countermeasure» Cm3-Privacy concerns
«Countermeasure» Cm2-Communication encryption
«System asset» Connection
-V4-Lack of communication
«Countermeasure» Cm1-Secure network service
isOn
*
1
«System asset» CLOUD
-V6-Insecure cloud interface
-V10-poor physical security
«Countermeasure» Cm9-Trust and governance
«System asset» Remote_Storage
-V3-Insecure networks service
1..*
isOn
1
«System asset» Local_Storage
{complete, overlapping}
isOn
0..*
1

Fig. 4. IoT security reference model

we introduce stereotype System asset to identify explicitly the component which potentially supports the data managed and controlled in the IoT system.

Characteristics of System Assets. As discussed in [6,13], vulnerability is *a characteristic of the system assets*. The vulnerabilities listed in Sect. 3.2 *characterise weaknesses of the system assets* presented in Fig. 3. We introduce these vulnerabilities as the attributes of the targeted vulnerable system assets.

For example, *Service* is vulnerable regarding insecure Web interface (V#1), insufficient authentication and/or authorisation (V#2), and insecure mobile interfaces (V#7). Vulnerability of insecure network services (V#3) could be found in the *network resources* and *remote storage*. A lack of communication encryptions (V#4) could potentially be considered in the *connection* and privacy concerns (V#5) should be considered when managing *IoT devices*. In the IoT systems, *cloud* plays an important role, thus, its interface should be considered regarding the insecure cloud interface (V#6) vulnerabilities. *IoT system* could be explored through the insufficient security configurability (V#8). As the *computing devise* is a part of the IoT system, its vulnerabilities regarding the insecure software and/or firmware (V#9) should be also taken into account. Finally, the poor physical security (V#10) could potentially open the gate for the attacker at the *data storage*, *computing device*, *IoT device* and *cloud*.

Countermeasures Becomes a Part of the IoT System. Security countermeasures are introduced to mitigate the security risks. In Fig. 4 we link the security countermeasures (see classes with stereotypes Countermeasure) to the system assets, which can be targeted by the security threat thus exploiting theirs vulnerabilities. Thus, these countermeasure should become a part of the IoT system (e.g., introduced as a part of the various IoT assets), thus reducing the potentiality of the security risk event happening.

Countermeasure on secure network services (Cm#1) mitigate risks with vulnerabilities of insecure network services (V#3), and communication encryption (Cm#2) – vulnerabilities related the lack of communication encryption (V#4). Countermeasures regarding the privacy concerns (Cm#3) help to mitigate security risks with vulnerabilities related to privacy concerns (V#5); secure software and/or firmware (Cm#4) – vulnerabilities related to insecure software and/or firmware (V#9). Countermeasure of physical security (Cm#5) address risks with vulnerabilities of poor physical security (V#10). Countermeasures to secure authentication and/or authorisation (Cm#6) mitigate risks with vulnerabilities of insufficient authentication and/or validation (V#2); to secure Web interface (Cm#7) – vulnerabilities of insecure Web interface (V#1); and to secure mobile interface (Cm#8) – vulnerabilities of insecure mobile interface (V#7). Countermeasures regarding the trust and governance (Cm#9) deal with the security risks with vulnerabilities of insecure cloud interface (V#6). Countermeasures regarding the fault tolerance (Cm#10) mitigate security risks with vulnerabilities of insufficient security configurability (V#8).

4 Analysis of Security Risks in IoT Systems

In this section we discuss few reported examples. The purpose is to illustrate the IoT assets, risks, and their vulnerabilities; potentially the countermeasures discussed in Sect. 3 should be applied to mitigate these risks.

Risk 1: An attacker with means to break the workflow of the popular Web services uses Mirai botnet to exploit the Linux kernel because it is out of date and users have not changed the default usernames and passwords on their devices. This leads to overload of the Websites servers overload, loss of Website reliability loss of routers, and negation confidentiality of the cameras' IPs [16].

Here the *business asset* is the cameras' IP supported by the Web services, Linux kernels, and other devices. The risk is possible because of the vulnerabilities V#2 (in Web service) and V#7 (in IoT device). It could be mitigated by a set of countermeasures selected from Cm#6 and Cm#8.

Risk 2: An attacker with the means to interrupt building heating process uses the distributed denial-of-service (DDoS) attack to target the unmonitored heating network. This leads to two unheated buildings during the freezing period, loss of heating controllers' reliability, and loss of confidentiality of data send over the heating network [12].

In this examples the *business asset* is the data sent over the network. This data is supported by (i.e., sent over) the heating network. The risk becomes possible because of the vulnerability V#3 in the network. This risk could be mitigated by countermeasures selected from Cm#1.

Risk 3: An attacker with means to slow down the workflow of the university servers uses Botnet and brute force password hack to target insufficiently monitored network activity and weak passwords. This leads to the harm to the system's network and servers and unreliable data [17].

The *business asset* is the data used in the university workflow. This data is supported by the university and protected using passwords. The risk becomes possible because of the vulnerabilities V#3 in the network (i.e., the University servers) and V#2 in the provided service (i.e., activity/workflow supported by the server). This risk could be mitigated by a set of countermeasures selected from Cm#1 and Cm#6.

Risk 4: An attacker uses a cryptovirology malware to target (allegedly) the insufficiently monitoring network in order to block access to, steal and publish to steal victim's data (until some ransom is paid). This leads to the harm to 70% of the city's CCTV systems [10].

In this example the *business asset* is the victim's data supported by the network. The risk becomes possible because of the vulnerabilities V#3 in the remote storage and V#8 in the IoT system. This risk could be mitigated by a set of countermeasures selected from Cm#1 and Cm#10.

Risk 5: Two white hat-hackers with the means to control someone's car remotely uses malicious software uploaded to the UConnect control module through insecure communication protocol (i.e., the UConnect module installed on the Chrysler car) because of the insecure GSM communication. This leads to the negation of the car's control commands, harm to the communication protocol, and loss of the cars control [7].

Here, the *business asset* is the car control commands supported by the UConnect module to sent them over the GSM communication channel. The risk becomes possible because of the vulnerability V#4 in the transport protocol (i.e., UConnect). This risk could be mitigated by countermeasures from Cm#2.

Risk 6: A white-hat attacker with the means to steal personal data and change data stored in the "My friend Cayla" doll's database because of the insecure Bluetooth connection. This leads to the loss of the data integrity in the "My friend Cayla" doll's database, unreliable Bluetooth communication and database of the "My friend Cayla" doll'. The risk provoke further impact resulting in the ban of the doll by German's government, qualified as an "espionage device" [15].

The *business asset* is a date supported by (sent through) the Bluetooth connection and (stored in) the doll's database. The risk becomes possible because of the vulnerabilities V#3 in the data storage (i.e., doll's database) and V#4 in the communication (i.e., Bluetooth communication). This risk could be mitigated by a set of countermeasures selected from Cm#1 and Cm#2.

5 Concluding Remarks

In this paper we have aligned the IoT system components to the ISSRM asset [6,13]. Then, following the OWASP [14], the vulnerabilities and countermeasures to mitigate them were highlighted. This potentially results in a reference model for securing IoT systems. We apply this initial reference model to the reported IoT security risks to illustrate instantiations of the IoT security risk concept.

The current reference model contains few limitations. Firstly, it basically covers the system assets and their vulnerabilities, but leaves the analysis of business assets (i.e., data exchanged in the IoT systems, business operations) and their security criteria aside. Regarding the security risk analysis, we have concentrated on the vulnerabilities, but further work is needed to highlight the profile of the threat agents, their attack methods, as well as the impacts on the IoT system and business assets. On the system countermeasure side, we make an assumption that to treat IoT security risk one takes risk reduction decision; however it is also important to understand consequence of other treatment decision (e.g., risk avoidance, retention or transfer). Finally, in our proposal we do not differentiate between the security requirements and controls.

In the future research we will also strengthen the proposed reference model with the definition of the explicit guidelines for the IoT asset, risk and risk countermeasure identification, as well as the method of the security trade-off analysis. We have also planned a hands-on case study where the relationship among the IoT assets, their vulnerabilities, and proposed countermeasures will be explored using the penetration testing.

Acknowledgement. This research has been supported by the Estonian Research Council (grant IUT20-55).

References

1. Abomhara, M., Koien.: Security and privacy in the Internet of Things: current status and open issues. In: International Conference on Privacy and Security in Mobile Systems (PRISMS). IEEE (2014)
2. Alabaa, F.A., Othma, M., Abaker, I., Hashem, I.A.T., Alotaibib, F.: Internet of Things security: a survey. J. Network Comput. Appl. **88**(15), 10–28 (2017)
3. Ammar, M., Russello, G., Crispo, B.: Internet of Things: a survey on the security of IoT frameworks. J. Inf. Secur. Appl. **38**, 8–27 (2018)
4. Banerjee, M., Lee, J., Choo, K.-K.R.: A blockchain future to Internet of Things security: a position paper. Digit. Commun. Networks (2018). https://doi.org/10.1016/j.dcan.2017.10.006
5. Bauer, M., Bui, N., De Loof, J., Magerkurth, C., Nettstrater, A., Stefa, J., Walewski, J.W.: Enabling Things to Talk. Springer, Heidelberg (2013)
6. Dubois, É., Heymans, P., Mayer, N., Matulevičius, R.: A systematic approach to define the domain of information system security risk management. In: Nurcan, S., Salinesi, C., Souveyet, C., Ralyté, J. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 289–306. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-12544-7_16
7. Greenberg, A.: Hackers Remotely Kill a Jeep on the Highway - with me in it (2015). https://www.wired.com/2015/07/hackers-remotely-kill-jeep-highway/
8. GSMA Connected Living. Understanding the Internet of Things (IoT) (2014)
9. Hellaoui, H., Koudil, M., Bouabdallah, A.: Energy-efficient mechanisms in security of the internet of things: a survey. Comput. Netw. **127**, 173–189 (2017)
10. Khandelwal, S.: Two Romanians Charged with Hacking Police CCTV Cameras Before Trump Inauguration (2017). https://thehackernews.com/2017/12/police-camera-hacking.html
11. Li, H., Zhou, X.: Study on security architecture for Internet of Things. In: Zeng, D. (ed.) ICAIC 2011. CCIS, vol. 224, pp. 404–411. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23214-5_53
12. Mathews, L.: Hackers Use DDoS Attack To Cut Heat To Apartments (2016). https://www.forbes.com/sites/leemathews/2016/11/07/ddos-attack-leaves-finnish-apartments-without-heat/
13. Matulevicius, R.: Fundamentals of Secure System Modelling. Springer International Publishing, Switzerland (2017). https://doi.org/10.1007/978-3-319-61717-6
14. OWASP. Welcome to OWASP. https://www.owasp.org/index.php/
15. Carolina. Goodbye Spy Toy: Germany Bans My Friend Cayla Doll (2017). https://www.hackread.com/good-bye-spying-toy-germany-bans-my-friend-cayla-doll/
16. The Guardian. DDoS attack that disrupted internet was largest of its kind in history, experts say (2016). https://www.theguardian.com/technology/2016/oct/26/ddos-attack-dyn-mirai-botnet
17. Weagle, S.: IoT-Driven Botnet Attacks US University. https://www.corero.com/blog/798-iot-driven-botnet-attacks-us-university.html
18. Yang, X., Li, Z., Geng, Z., Zhang, H.: A multi-layer security model for internet of things. In: Wang, Y., Zhang, X. (eds.) IOT 2012. CCIS, vol. 312, pp. 388–393. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-32427-7_54

Information Requirements for Big Data Projects: A Review of State-of-the-Art Approaches

Natalija Kozmina(✉), Laila Niedrite, and Janis Zemnickis

Faculty of Computing, University of Latvia, Raina Blvd. 19, Riga, Latvia
{natalija.kozmina, laila.niedrite}@lu.lv,
janiszemnickis@gmail.com

Abstract. Big data technologies are rapidly gaining popularity and become widely used, thus, making the choice of developing methodologies including the approaches for requirements analysis more acute. There is a position that in the context of the Data Warehousing (DW), similar to other Decision Support Systems (DSS) technologies, defining information requirements (IR) can increase the chances of the project to be successful with its goals achieved. This way, it is important to examine this subject in the context of Big data due to the lack of research in the field of Big data requirements analysis. This paper gives an overview of the existing methods associated with Big data technologies and requirements analysis, and provides an evaluation by three types of criteria: (i) general characteristics, (ii) requirements analysis related, and (iii) Big data technologies related criteria. We summarize on the requirements analysis process in Big data projects, and explore solutions on how to (semi-) automate requirements engineering phases.

Keywords: Big data · Requirement analysis · Literature review

1 Introduction

Big data usually is understood as large and complex data that needs new computer technologies and new methods for processing. The large amount of data is not the only property that defines the necessity for new techniques. The most popular definition of Big data [1] includes 3 Vs: Volume, Variety, and Velocity. These Vs characterize the large amount and the complexity of data as well as the data generation speed that causes the necessity to deal with the streaming data. Other Vs were added later to the list, for example, veracity and value, which are describing the uncertainty and business value of Big data.

Big data is involved in many human everyday activities, it is also connected with many other widely known terms e.g. social networks, Internet of Things (IoT), Big data analytics, cloud computing, NoSQL, etc. that help to understand how the Big data is emerging or how it can be processed.

Currently many attempts are made to develop Big data solutions having different level of success. Projects' failures are often caused by the problems [2] that are

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 73–89, 2018.
https://doi.org/10.1007/978-3-319-97571-9_8

determined by the Big data features e.g. finding the best way how to extract the value from Big data, integration problems of Big data sources, or data quality problems. Researches provide also more technological Big data issues [3]: storage and processing issues, analytical, and technical challenges.

Problems of Big data projects' failures [4] can be solved by the right choice of tools and methods, for example, automation and agile techniques [5]. Many proposals solve some particular aspect of the whole development process of a Big data solution, but the questions remain the same: (a) how to design it methodologically starting from the business needs? (b) what is the appropriate data to be collected according to the *information requirements* (IR) of the company? IR define information, which should be available after the development of the information system is finished.

This paper gives an overview of the existing methods associated with Big data technology and requirements analysis, and provides an evaluation by three types of criteria: (i) general characteristics, (ii) requirements analysis related, and (iii) Big data technologies related criteria.

The structure of the paper is as follows: Sect. 2 provides the description of the literature review process, in Sect. 3 we elaborate on the review results and categorize them, and Sect. 4 finalizes the paper with conclusions and discussion.

2 The Literature Review Process

In this section we describe the literature review process that was conducted according to the most commonly used guidelines given by Kitchenham and Charters [6]. The literature review process is composed of three major phases: (i) preparing for the review, (ii) conducting the review process, and (iii) reporting the results of the study.

2.1 Preparing for the Review

Setting the Goal. The goal of our literature review was to explore the aspects of information requirements analysis in the context of Big data.

Research Questions. We base our study on the following research questions:

RQ1. How the requirement analysis applied in the context of Big data?
RQ2. What empirical methods have been applied for the requirement analysis in the field of Big data?
RQ3. Is it feasible to generate the information requirements (IR) in a Big data project by processing the existing data in a (semi-) automatic way?

2.2 Conducting the Review Process

Source Selection Strategy and Search Queries. We performed search in 6 widely used electronic publication databases: ACM, Scopus, IEEE Xplore, SpringerLink, Web of Science, and Google Scholar.

A Google Scholar search query was as follows: <*"Requirement engineering" AND "Big Data" AND ("Analysis Requirements" OR "Information Requirements" OR*

"Data Requirements")> . Search queries for other databases could slightly differ due to peculiarities of each particular database, but the semantic meaning stayed the same.

Inclusion (IC) and Exclusion Criteria (EC). A study was selected for further detailed analysis, if it met all the IC and did not cover any of the EC. All the IC and EC are summarized in Table 1. These criteria would ensure that the results of the survey are the most relevant and in line with research questions stated in Sect. 2.1.

Table 1. Inclusion (IC) and exclusion (EC) criteria applied at a certain analysis stage

ID	*Criteria description*	*Stage*
IC1	The publication date falls into the time interval from 2014 to 2017	1
IC2	The study is indexed in at least one of the selected publication databases	1
IC3	The language of the study is English	1
IC4	The title, keywords, and abstract of the study is related to at least one of the formulated research questions	1
IC5	The contents (headings, figures, table, introduction, and conclusions) of the study is related to at least one of the formulated research questions	2
EC1	A duplicate paper	2
EC2	In case of multiple versions of the paper, only one is included (either the latest or the fullest)	2
EC3	The study is a (PhD, Master, or Bachelor) thesis, a survey or a literature review, a poster paper, a standard, a books, a tutorial	2
EC4	The keyword "Big data" is not present in the paper body (e.g. in the Reference section only)	3
EC5	The approach presented in the study is too specific and cannot be generalized to be applied in other domains (e.g. a study on geospatial data or electromagnetic inference)	3

Paper Selection and Search Results in Numbers. The process of paper selection included 3 stages explained further. Moreover, during all 3 stages we subsequently applied IC and EC (see Table 1) to the list of literature studies.

Stage 1: Skimming the paper title, keywords, and reading the abstract. The overall number of related papers after searching in 6 electronic publication databases was 242. Eliminating the irrelevant studies during the quick scan resulted in 75 papers.

Stage 2: Reading the introduction, conclusions, section and subsection headings, tables, and figures. The subsequent analysis included fragmental reading of the papers. The studies that were out of scope (i.e. the topic of Big data technologies and/or requirements analysis was not developed) or irrelevant according to IC and EC were excluded leaving the 26 papers for further examination.

Stage 3: Thoroughly reading the full-text of the paper. During the final phase of the selection process, we evaluated each paper by quality assessment criteria (see the subsection below) in addition to the IC and EC.

Quality Assessment Criteria. We have applied the following criteria to ensure the quality and relevance of the studies complies with the goal of our research:

Q1. Is the contribution of the paper related to requirement analysis in Big data?
Q2. Is the contribution of the paper novel and it covers any improved methodology or open issue?
Q3. Is there any case study or approbation example for the methodology proposed in the paper?
Q4. Is the theoretical approach or approbation example stated clearly?

Each study received an assessment of Q1–Q4 as a numeric value in the range between 0 (strongly disagree) and 5 (strongly agree). We have excluded the studies that received lowest scores (i.e. 0 or 1) in at least 2 quality assessment criteria. The final set of papers consisted of 22 studies; their detailed classification is given in Sect. 3.

Limitations of our study are dictated by the choice of the electronic publication databases, formulated search queries, and correspondence to IC, EC, and Q1–Q4.

3 Classification of the Review Results

In this section we present an overview of the studies selected for an in-depth analysis. We summarize the main features of each paper in Tables 2, 3 and 4. We elaborate on the results of our literature study by providing answers to the stated research questions.

Table 2. General characteristics

	Type of the contributions	*LoA*	*Motivation for research*	*D-I*
[7]	DLC model, use cases	high	the absence of a general DLC model to easily adapt to a different/new scenario	yes
[8]	application scenarios	med	3 main challenges require consolidated solutions: data storage, data analytics, and data integration	no
[9]	guidelines (on structure of a Hive data model), empirical study	high	the lack of methodological guidelines for defining the structure of the tables in Hive	yes
[10]	methodology (to create the model), application scenario	med	the absence of the RE artefact models to support RE process design and project understanding	yes
[11]	case study	low	no research that examined GORE method/framework in Big data software development	no
[4]	methodology, case study	high	the lack of methods to manage, analyze, and visualize Big data systematically	yes

(*continued*)

Table 2. (*continued*)

	Type of the contributions	*LoA*	*Motivation for research*	*D-I*
[12]	methodology, survey	med	insufficient early involvement of users and stakeholders for better understanding of project goals	yes
[13]	framework, case study	low	intensive disk I/O operations and focus on optimizing individual analysis tasks are the biggest bottleneck of existing genomics analysis pipelines	no
[14]	framework, empirical study	med	the absence of works to integrate phrase extraction and phrasal segmentation	yes
[15]	methodology	high	improvement of the classical NLP techniques to be able work with Big data	yes
[16]	methodology, empirical study	low	improvement of ETL process with mapReduce technologies	yes
[17]	methodology	high	development of Big data as SOS technology	yes
[18]	methodology	low	discovery of the IS processes with Semantic web technologies	yes
[19]	case study (challenges and examples)	med	development of the context model for Big data software engineering	yes
[20]	case study	med	companies don't use all available data to improve factoring	yes
[21]	empirical study	low	development of the Ophidia framework to support (meta-)data management	yes
[22]	methodology (requirement-driven approach for Big data application services)	med	the lack of requirement engineering guidelines to develop Big data services	yes
[23]	methodology for the regeneration of the user interfaces for component based Web applications at runtime	med	necessity for component-based user interfaces to be intelligent and evolve over time	no
[24]	pattern-driven analysis requirements modeling method	high	difficulties to describe and understand the analysis problems due to their abstract nature	yes
[25]	a tool for automated web mining and Big data analysis to generate training data for supervised architecture-traceability techniques	med	creating and maintaining traceability links between requirements, architecture, and source code is costly and complicated	yes

(*continued*)

Table 2. (*continued*)

	Type of the contributions	*LoA*	*Motivation for research*	*D-I*
[26]	architecture of a system	low	the necessity of execution of different NLP applications using parallel computing methods	yes
[27]	method for building specifications of systems based on the data sets which offer system requirement information	high	rapidly growing data sets provide the possibility to extract useful information including the information about systems' requirements	no

3.1 Evaluation of the Selected Studies

Table 2 provides a full list of analyzed papers and their general features:

- Type of the contributions in the paper (can be more than one category): methodology, empirical study, case study, guidelines, etc.;
- Level of abstraction (*LoA*): high, medium (*med*), or low;
- Motivation for research: the main reason for conducting the study;
- Domain-independent (*D-I*): yes or no.

Table 3 includes only studies that cover aspects of the requirement engineering, and are characterized according to the following criteria:

- Requirement artifacts: goals, scenarios, solution-oriented, etc.;
- Requirement development (*RD*) activities in focus (can be more than one category): elicitation, analysis, specification, validation, not specified (*N/A*);
- Requirement processing techniques: an outline of the proposed methods;
- (Semi-) Automatization capabilities (*S/A cap.*): high, medium (*med*), low, or not specified (*N/A*).

Table 3. Requirements-related aspects

	Requirement artifacts	*RD activities in focus*	*Requirement processing techniques*	*S/A cap.*
[7]	scenarios (applicability of the DLC in different projects)	all (no details)	direct/indirect data collection from sources, managing the ranges of sources, exploring and discovering new sources	N/A
[9]	solution-oriented (MD data model to tabular form)	elicitation, analysis	MD data models are constructed in compliance with analytical requirements defined by users	high
[10]	goals, scenarios (showing the capabilities of RE artefact model)	elicitation, specification	requirements, constraints, and scenarios gathered during interviews are grouped according to the classes in the artefact model	low

(*continued*)

Table 3. (*continued*)

	Requirement artifacts	*RD activities in focus*	*Requirement processing techniques*	*S/A cap.*
[11]	goals	elicitation, specification	requirements generation from goal-oriented models (i* and KAOS)	low
[4]	goals	all	queries are composed according to the IR; the source data is queried for availability; the raw data is transformed into facts and dimensions; MD data models of each data source are iteratively integrated into one	med
[12]	goals	elicitation, analysis	elicitation of the requirements on the BDA-as-a-service, construction of the Kano Questionnaire (KQ) for their classification, KQ analysis	med
[17]	requirements model	elicitation, analysis	modeling requirements from user needs and "BigData 7 V"	low
[19]	scenarios	all	eliciting behavioral scenarios of the desirable system responses	low
[22]	solution-oriented (service requirements)	elicitation, analysis	defining new requirements based on experience and iterative implementation of them according to user and business needs	low
[23]	solution-oriented (for creating evolutionary mashup user interfaces)	all (no details)	capturing and storing user interaction data, applying ML algorithms to analyze interaction data, model transformation methods to get interface conversion rules	low
[24]	goals, analysis patterns	all (no details)	analysis patterns are used, analysis requirements are modeled	low
[25]	solution-oriented (requirements traceability support)	validation	architectural choices known as tactics are used, web mining methods, Big data analysis techniques	high
[27]	solution-oriented (requirements formal model)	specification	the data is restricted to the sequence of actions that the system behave, the models are the abstract automata	N/A

Table 4 includes interesting aspects from the field of Big data such as:

- Applicability of requirement development (*RD*) activities in Big data context: high, medium (*med*), low, not specified (*N/A*);
- Structured/Unstructured data processing capabilities (*S/U*): structured (*S*), unstructured (*U*), both, or not specified (*N/A*);
- Data processing techniques: an outline of the proposed methods if any;
- V-characteristics (*Vs*): (varies from 3Vs to 7 Vs or not specified (*N/A*)).

Table 4. Big data related aspects

	Applicability of RD activities in Big data context	*S/U*	*Data processing techniques*	*Vs*
[7]	high (fits the context of Big data well and is adjustable to any scenario to manage any kind of requirements keeping the high level of data quality)	both	transformations, quality check, pre-processing, post-processing	6Vs: Volume, Variety, Velocity, Value, Veracity, Variability
[9]	high (covers the aspects of integration of the multidimensional data sources into the Hadoop lifecycle)	U	automatic rule-based transformation of a MD data model into a Hive tabular schema	N/A
[10]	high (Big data requirements and scenarios are separate classes of the artefact model)	both	N/A	3Vs: Volume, Variety, Velocity
[11]	med (4 general requirements for Big data application were modeled as softgoals)	both	transformations, quality check	4Vs: Volume, Variety, Velocity, Veracity
[4]	high (covers management, analysis, and visualization of Big Data)	both	data stages definition, data sources acquisition and management, adding value to the data, implementation of a BDW, visualizations for Big Data	5Vs: Volume, Variety, Velocity, Value, Veracity
[12]	high (the framework includes customizable models of the Big Data Analytics process and its artifacts)	both	summarization and graphical representation of Kano questionnaires' results	N/A
[13]	N/A	U	genomics data acquisition and parsing, ETL processes, pre-processing, analysis, visualization	N/A
[14]	med (unstructured text can be transformed into structured units)	U	quality phrase mining, NLP	N/A
[15]	N/A	U	data mining, NLP	N/A
[16]	N/A	S	ETL, mapReduce	N/A

(*continued*)

Table 4. (*continued*)

	Applicability of RD activities in Big data context	*S/U*	*Data processing techniques*	*Vs*
[17]	high (method describes Big data specific requirements)	N/A	system of systems (SoS)	7Vs: Volume, Velocity, Variety, Veracity, Value, Variability and Visualization
[18]	med (using the semantic technologies in Big data file system to discover IR)	S	semantic Web technologies	N/A
[19]	high	both	Multi-Peak, granular computing	4Vs: Velocity, Volume, Variety, Veracity
[20]	N/A	both	Simulations	N/A
[21]	N/A	both	Big data technologies	N/A
[22]	high (Big data application service selection based on requirements catalog)	both	depend on selected services combined in the service pipelines for Big data processes	N/A
[23]	high (user interaction data processing with Big data technologies to get new requirements for user interface evolution)	both	data storage, view definition, transformations are performed by ML algorithms	N/A
[24]	high (no details specified)	N/A	only theoretical model is provided	N/A
[25]	high (dataset generation for traceability techniques from Big data sources)	both	document indexing techniques for indexing and searching	N/A
[26]	N/A	U	Map/Reduce, TF-IDF relevance function for keywords extraction	N/A
[27]	N/A	N/A	Big data as a data source to represent the systems' behaviour and requirements	N/A

3.2 Evidence of the Requirement Development Activities in Big Data Projects

We have united the RQ1 and RQ2, since most of the corresponding papers cover both a theoretical approach and an empirical study. To analyze the evidence of the requirements development activities in Big data projects, we have taken the intersection of the studies included in Tables 3 and 4, which resulted in 13 papers. We grouped them by the column “Requirement development activities in focus” of the Table 3. In total, there are 5 groups: (i) all (elicitation, analysis, specification, and validation), (ii) elicitation and analysis, (iii) elicitation and specification, (iv) specification, and (v) validation. Let’s consider the contributions that fall into each of the groups.

Group (i): All Requirement Development Activities. The central figure of the study [7] is an advanced Data LifeCycle (DLC) model. Its purpose is to define the sequence of phases in the data life, specify management policies for each phase, and describe the relationship among them. The Comprehensive Scenario Agnostic Data LifeCycle (called COSA-DLC) presented in [7] consists of 3 interconnected blocks: Data Acquisition, Data Processing, and Data Preservation. The Data Acquisition block consists of the 4 phases: Data Collection (acquiring data from all sources and devices according to the business requirements), Data Filtering (basic data transformations for optimization of the volume of data between the Collection and Quality phases), Data Quality (rejecting the data of low quality) and Data Description. The Data Processing block includes 3 phases, namely Data Process (pre- or post-processing of (raw) data into a more sophisticated form in compliance with business requirements), Data Quality (performing checks before storing the data), and Data Analysis. Finally, the Data Preservation block is ensures the data storage performing any required actions for data classification and making the data available for the future processing.

The main idea in [7] was to suggest the activities to be performed in each of the blocks of the model (i.e. *what?*) instead of focusing on details (i.e. *how?*), thus, it is hard to make conclusions on the methods applied to ensure that the collected data conforms to business requirements. The empirical validation of the approach is reflected in a very brief description of the 2 use cases (i.e. smart city and scientific library) to demonstrate how the model can be adapted in different scenarios.

An iterative methodology [4] is presented that in a systematic way improves the management, analysis, and visualization of Big Data taking into account the best practices. There are 5 phases of the methodology: (1) data stages definition, (2) data sources acquisition and management, (3) adding value to the data, (4) selection and implementation of a Big Data Warehouse (BDW) and, finally, (5) development of visualizations for Big Data.

In terms of our study stages (1) and (3) are of particular interest. During stage (1) information and non-functional requirements are identified for each analysis task in the project. Information requirements serve to determine the data structure, whereas non-functional requirements (e.g. time constraints, data quality requirements, query latency) detect the most appropriate tools and models for each analysis task. The authors apply multidimensional (MD) modeling technique, because it is quite simple and efficient. The stage (3) starts with the exploration of raw data sources. For each IR (from stage 1), the source data are queried to check its availability and potential usefulness. Then, the raw data will be converted into facts and dimensions. MD models of each data source are iteratively integrated into one model using the similarity between their facts. Lastly, an enrichment of the modeled BDW is performed. Apache Pig and/or Hive can be used to query raw data according to the MD schema of the BDW. This process can be repeated continuously, if needed. A case study was performed on the analysis of electricity consumption data produced by smart meters distributed worldwide.

In [19] a Big data software engineering contextual model is generated. The authors consider requirements engineering as one of the major Big data challenges. Thus, they suggest creating domain models, eliciting use-case scenarios and requirements from stakeholders and other sources, developing functional and behavioral models, performing analysis, prioritization, and validation. While designing the system, specific characteristics of the Big data such as volume, velocity, variety should be included, as well as domain-specific requirements i.e. security, privacy, reliability, ease of use, etc.

A method [23] that allows the component-based interfaces being up-to-date according to the changing user requirements is proposed. The requirements can change over time in distributed environments, when new users or components appear. The methodology consists of 5 steps (S1) – (S5). (S1) captures information about users interaction (users characteristics and operations performed in the interface), environment (time, location and others), and interface status after interaction. The captured interaction data has large volume, data is heterogeneous and is generated at high speed, and therefore data is stored using Big data techniques. (S2) is to define different views on the data captured in (S1). The goal is to select appropriate data as entry elements for different ML algorithms. (S3) is intended for application of ML algorithms to test and evaluate the right algorithm to extract valid knowledge from interaction data. Valid knowledge in this case allows generating new rules for interface improvements. The algorithms of interest are: Clustering, Association Learning, Neural Networks, etc. (S4) uses model transformations to transform the outputs of ML algorithms in the conversion rules that can be applied to user interface depending on domains. (S5) provides evaluation rules for the new generated conversion rules from (S4). A case study is a component-based graphic user interface ENIA for environmental management.

An analysis requirements modeling method [24] is based on analysis patterns reusing the previous experience to elicit and model analysis requirements by documenting the problem, goals, and analysis models. The description of an analysis pattern is based on a template. The method consists of 5 phases that allow eliciting analysis requirements. (1) From the given Initial User Problem Diagram (IUPD) the analysis requirements are extracted together with related goals, domains, and machines to generate an initial Analyst Problem Diagram (APD). (2) Appropriate analysis pattern (AP) is selected from the AP repository and applied to the IAPD, and the refined APD is prepared. (3) Best analysis model is selected from the model repository. (4) The analysis model is configured to get an analysis machine (AM). The AM is composed into IUPD. The proposed method uses not only the experience of the analyst but also experiences stored in repositories.

Group (ii): Elicitation and Analysis. An advantage of the approach [9] is considered to be the absence of the necessity to specify IR after the MD model is defined. A set of rules to transform automatically an existing MD data model into a tabular schema that can be implemented in Hive and queried with HiveQL is proposed. More precisely, star or constellation schema can be represented as a set of tables based on dimensional lattice, identification of descriptive and analytical column groups, association of column groups to tables, identification of columns in each table, and identification of partitions and buckets. As a final step, data analysis and visualization tools (e.g. Tableau) can be applied to for visual representation of data. A small-scale empirical study demonstrates the approach applicability with 2 fact tables and 4 dimensions. After identification of the dimension lattice, the resulting set includes 11 tables with column groups that can be implemented in Hive.

Ardagna et al. [12] report the results achieved by TOREADOR framework by actively involving users and stakeholders in the requirement elicitation phase with an objective to prioritize the requirements at the very early stage, since it is an important aspect for Big data projects. An initial list of requirements based on the contribution of each stakeholder is created. All the partners express the elicited requirements using a uniform specification format: Name/Property/Rationale/Scope, Source, and Target/Priority/Dependencies. The requirements are organized into the 6 categories: Preliminaries, Model, Infrastructure, SLA (service level agreement), Legal, Pilot specific requirements. The output of the requirements elicitation phase is a consistent and structured dataset of stakeholders' requirements. The empirical study involves conducting a survey (Kano questionnaire or KQ), which results in a Kano model after processing and aggregating the results. The Kano model classifies requirements based on their position along two dimensions, namely, the degree of satisfaction and the level of functionality, this way, ensuring the prioritization of the requirements. The KQ contains a list of question pairs for each product requirement. In its turn, the question pair is composed of a functional question (i.e. how would one feel if a requirement is fulfilled), and a dysfunctional question (i.e. how would one feel if a product fails to achieve the requirement). The results of the KQ are summarized according to the categories that emerge after the application of the Kano Evaluation Table.

The paper [22] focuses on requirements processing for Big data application services, providing a requirement driven approach for development of services and for support of services runtime environment. The services computing in the Big data context means development of service pipelines to support processing of Big data volumes. The whole Big data lifecycle is covered including data collection, storage, and analysis as well as the service development process, including implementation, improvement, and optimization. When appropriate, the approach suggests the usage of existing services that correspond to the business needs.

The authors provide requirements classification parameters that they use as a basis for the Big data Services Requirements Catalogue that cover following aspects: requirements type, data type, business process, and performance index. Authors provide examples of requirements related to components selection: type of storage, data consistency management strategy, data arrival mode, etc.

The proposed approach how to build a Big data processing system that is aligned to the business needs and consists of appropriate components and services, and that is afterwards iteratively improved according the new requirements gathered from log files of used services. The authors use data mining methods to select from the requirement repository appropriate services according to the elicited features.

The method is applied in the health care domain (eHealth and industry) to illustrate the proposed approach for service requirements processing.

Group (iii): Elicitation and Specification. The recent study [10] was driven by the absence of the Requirement Engineering (RE) artefact models that would facilitate the design of RE processes and would enhance the common understanding of the Big data software projects. The paper sheds light on post-processing of elicited, analyzed, prioritized, and specified requirements, and proposes a RE artefact model for Big data end-user applications (called BD-REAM) and a method to create it. The RE artefact model requires 4 steps to be developed: (i) elements and concepts, (ii) artefact relationships, and (iii) cardinalities are sequentially defined, and as a final step, (iv) the artefact model is compiled. The acquired model includes 22 artefacts and a number of inter-relationships that cover IR oriented classes (e.g. data source, data transformation, quality requirements), and Big data requirements and scenarios that aim at the usage of the system. Currently no automated approach to translate scenarios into formulated system requirements was proposed meaning that an analyst would do it manually. The application scenario shows the capabilities of the designed RE artifact model while developing a Big data project for a financial services company.

A generic requirement model [11] for Big Data application is proposed. While being supplemented with Big data related classes, the model mostly makes use of the existing i* Framework and Knowledge Acquisition autOmated Specification (KAOS). The i* models take advantage of the dependencies among actors. Meanwhile, KAOS is a multi-paradigm that allows a combination of different levels of expression and rationale: semiformal for modeling and structuring goals, qualitative for alternative selections and formal for the critical elements. KAOS consists of 4 models: goal, responsibility, object, and operational model. The 4 general requirements for Big data applications are defined based on Big data 4Vs characteristics and challenges: (i) huge database capacity, (ii) fine database performance, (iii) quality and structure of data, and (iv) guaranteed privacy and security of data. Requirement modeling for Big data application using i* involves 3 actors: Big data application, user, and application controller, and is represented by 2 models: Strategic Dependency (SD) and Strategic Rationale (SR). SR is a more elaborated version of SD that includes tasks to be fulfilled to satisfy the softgoals. Requirement modeling for Big data applications with KAOS suggests that each goal is derived based on the question "why" and "how". The evaluation of applying both approaches for the development of the Big data application for a government agency is described very generally and briefly resulting in 26 functional and 10 non-functional requirements approved by a stakeholder.

The authors [17] claim that Big data can not be interpreted as a regular system; instead it should be treated as a System of Systems (SoS). They propose to adapt the SoS to Big data using a specific method. It forms a model from all the collected data in 3 stages: (1) identification of the requirements from two aspects of user requirements and requirements related to the Big data constraints, (2) definition of criteria for each requirement, which serves as a basis for the information obtained in stage (1), and (3) modeling the requirements that includes the detailed description of how the requirements are interconnected with the Big data definition.

Group (iv): Specification. The authors [27] propose a software design method based on requirement data collected from users and describe the systems' behaviour. The requirement data is analysed in order to get formal specifications of systems.

The authors consider only the type of data that are sequence of actions of the systems and the models are abstract automata. The method is applied in a case study for a Spatial-Temporal System.

Group (v): Validation. Santos et al. [25] describe techniques are based on supervised machine learning algorithms, which allow tracing links between requirements, architecture, and source code. The authors do not provide new methods, but show how by means of automated tool usage that is based on existing web mining and Big data analysis methods can be solved problem with dataset generation for supervised architecture-traceability techniques. A BUDGET tool, which supports researchers in software architecture and requirements engineering fields, is proposed. The goal of the BUDGET tool is dataset generation for traceability techniques.

3.3 Evidence of the (Semi-) Automatization Capabilities

Another objective of our study was to reveal possible (semi-) automatization of IR. To provide answers to the RQ3, we have selected the studies with "high" or "medium" values of the column "(Semi-) Automatization capabilities" in Table 3.

High-level (Semi-) Automatization. Santos and Costa [9] introduce a set of rules that automatically transform a MD data model into a tabular schema in Hive. First, the dimensional lattice that incorporates dimensions and all the combinations between dimensions is generated. Then, column groups (descriptive for attributes and analytical for measures or business indicators) are detected. Next, another rules allow to associate column groups of both types to physical Hive tables, and define aggregation functions. Then, columns in Hive tables are classified either as descriptive or analytical depending on the column group they originate from. Lastly, another set of rules ensures partitioning and bucketing.

The goal of the BUDGET tool in [25] is dataset generation for traceability techniques; therefore, the tool has following features. (1) Open source systems code repository (from GitHub, SourceForge, Apache and Google Code). (2) The web-mining component that uses Google Search API to search technical specifications of tactics in different technical libraries (e.g. MSDN). The authors define tactics as building blocks of the software architecture, which satisfy some quality requirements. (3) An automated Big-data analysis engine that uses repository from (1) to extract sample implementations of tactics. (4) The tool supports different data-sampling strategies (stratified and random sampling techniques). (5) Filtering feature allows to get more targeted results, e.g. in specific programming language.

Medium-Level (Semi-) Automatization. Tardio et al. [4] automatically determine the availability and usefulness of the source data; sources are queried with tools like Apache Pig or Hive, while queries are formed in compliance with the IR. Then, the raw data is translated into facts and dimensions according to the results of querying. Unfortunately, there is no suggestion in [4] on how to automate generation of queries to source data based on requirements, or how to distinguish IR stated for facts, measures, or dimensions. A semi-automated approach is applied at the integration stage: fact similarities of each MD data model are calculated, similar MD data models are grouped, a MD data model is created for each group, and finally, to integrate multiple models into one, remaining MD elements are compared in a non-automated way.

An approach for requirement prioritization put forward in [12] could be characterized as semi-automatic. It includes the composition of the Kano Questionnaire according to the list of elicited requirements. The Kano Questionnaire in fact is a survey that includes functional and dysfunctional questions with multiple-choice answers that are mapped to numeric values. For evaluation of each requirement, for example, assigning Functional (F), Dysfunctional (D), and Importance (I) scores can be applied that are calculated by corresponding formulae, while I is an average importance value.

4 Conclusions and Discussion

Summarizing the studies on Big data and requirements engineering, the importance of defining requirements while developing a Big data solution is indisputable. One of the reasons for Big data project failures is that not all the necessary information requirements are provided initially or the user's expectations are different (e.g. regarding data quality, data access, etc.). In some cases, the failure can be explained by the fact that no analytical steps had been carried out before the development.

We conclude that the requirements and data analysis should be integrated in 2 phases of development of the Big data solution. (1) Prior to the development it is possible to identify the user expectations timely and compare them with available resources, source data, quality, granularity, and available resources of the project i.e. budget, time, skills, and environment (e.g. as in [4, 10, 11, 17]). (2) When the data has already been loaded from the source systems, a user may not be able to define all IR. However, when all the data is collected, new relationships and values can be explored (e.g. as in [9, 14, 15]).

We suggest using existing data mining techniques to discover new information and to offer it to users as possible information requirements. The challenge is to integrate existing data mining methods into the Big data ecosystem. In addition, during our study we have discovered approaches that have not been directly applied for requirements analysis, however, we consider the certain techniques useful at the stage of (semi-) automatic information requirements processing.

As stated in [15], the existing word processing technologies are not capable of handling large amounts of data. The approach [15] allows getting a deeper insight into the document contents when processing it with data mining techniques. The solution improves existing data mining techniques by the algorithms based on ontological domain modeling, NLP, and ML. Ontology is being crafted to simulate a particular domain, while data mining technologies ensure automatic data retrieval from (un) structured data. NLP identifies new terms such as persons or relations. Cheptsov et al. [15] describe 2 examples of word processing techniques that couldn't have been feasible to implement with the existing tools available.

Nesi et al. [26] propose a system that allows execution of different NLP applications that is based on open source GATE APIs and implemented via MapReduce on a multi-node Hadoop cluster. The architecture and implementation of the system is validated by a specific design case for keywords and keyphrases extraction from unstructured text. The unstructured text is gathered from the web by crawling phase and is processed later by keyword/keyphrases extraction process. The main task of the application is the relevance estimation of keywords and keyphrases within the domains' corresponding document set done by computing TF-IDF relevance function.

Liu et al. [14] present a framework that extracts quality phrases from text corpora integrated with phrasal segmentation, thus, transforming unstructured text into structured units. The framework requires only limited training but the quality of phrases generated is close to human judgment. Efficiently and accurately extracting quality phrases is the main goal of the study. The full procedure of phrase mining is as follows. (1) Generate frequent phrase candidates according to popularity requirement. (2) Estimate phrase quality based on features about concordance and informativeness requirements. (3) Estimate rectified frequency via phrasal segmentation. (4) Add segmentation-based features derived from rectified frequency into the feature set of phrase quality classifier; repeat step (2) and (3). (5) Filter phrases with low rectified frequencies to satisfy the completeness requirement.

We consider that NLP technologies mentioned in the findings described above are suitable to be integrated into elicitation of requirements and their post-processing.

References

1. Beyer, M.A., Laney, D.: The importance of 'big data': a definition. Gartner, Stamford (2012)
2. Kart, L., Heudecker, N., Buytendijk, F.: Survey Analysis: Big Data Adoption in 2013 Shows Substance Behind the Hype. Gartner Inc. (2013)
3. Katal, A., Wazid, M., Goudar, R.H.: Big data: issues, challenges, tools and good practices. In: IC3 2013, pp. 404–409. IEEE Press (2013)
4. Tardio, R., Mate, A., Trujillo, J.: An iterative methodology for big data management, analysis and visualization. IEEE BigData 2015, pp. 545–550 (2015)
5. Di Tria, F., Lefons, E., Tangorra, F.: Design process for big data warehouses. In: DSAA 2014, pp. 512–518 (2014)
6. Kitchenham, B., Charters, S.: Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical report. Keele University (2007)
7. Sinaeepourfard, A., Garcia, J., Masip-Bruin, X., et al.: Towards a comprehensive data lifecycle model for big data environments. In: BDCAT 2016, pp. 100–106. ACM, New York (2016)
8. Caldarola, E.G., Picariello, A., Castelluccia, D.: Modern enterprises in the bubble: why big data matters. SIGSOFT Softw. Eng. Notes **40**(1), 1–4 (2015)
9. Santos, M.Y., Costa, C.: Data warehousing in big data: from multidimensional to tabular data models. In: C3S2E 2016, pp. 51–60. ACM, New York (2016)
10. Arruda, D., Madhavji, N.H.: Towards a requirements engineering artefact model in the context of big data software development projects: Research in progress. IEEE Big Data 2017, pp. 2314–2319 (2017)

11. Eridaputra, H., Hendradjaya, B., Sunindyo, W.D.: Modeling the requirements for big data application using goal oriented approach. In: ICODSE’14 (2015)
12. Ardagna, C.A., Ceravolo, P., Cota, et al.: What are my users looking for when preparing a big data campaign. IEEE BigData Congress 2017, pp. 201–208 (2017)
13. Abdullah, T., Ahmet, A.: Genomics analyser: a big data framework for analysing genomics data. In: BDCAT 2017, pp. 189–197. ACM, New York (2017)
14. Liu, J., Shang, J., Wang, C., et al.: Mining quality phrases from massive text corpora. In: SIGMOD 2015, pp. 1729–1744 (2015)
15. Cheptsov, A., Tenschert, A., Schmidt, P., Glimm, B., Matthesius, M., Liebig, T.: Introducing a new scalable data-as-a-service cloud platform for enriching traditional text mining techniques by integrating ontology modelling and natural language processing. In: Huang, Z., Liu, C., He, J., Huang, G. (eds.) WISE 2013. LNCS, vol. 8182, pp. 62–74. Springer, Heidelberg (2014). https://doi.org/10.1007/978-3-642-54370-8_6
16. Mallek, H., Ghozzi, F., Teste, O., Gargouri, F.: BigDimETL: ETL for multidimensional big data. In: Madureira, A.M., Abraham, A., Gamboa, D., Novais, P. (eds.) ISDA 2016. AISC, vol. 557, pp. 935–944. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-53480-0_92
17. Tikito, I., Souissi, N.: Data collect requirements model. In: BDCA 2017, 7 p. ACM, New York (2017). Article 4
18. Di Francescomarino, C., et al.: Semantic-based process analysis. In: Mika, P., et al. (eds.) ISWC 2014. LNCS, vol. 8797, pp. 228–243. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-11915-1_15
19. Madhavji, N.H., Miranskyy, A., Kontogiannis, K.: Big picture of big data software engineering: with example research challenges. In: BIGDSE 2015, pp. 11–14. IEEE Press (2015)
20. Shao, G., Shin, S., Jain, S.: Data analytics using simulation for smart manufacturing. In: Proceedings of the Winter Simulation Conference, pp. 2192–2203. IEEE Press (2014)
21. Fiore, S., et al.: Big data analytics on large-scale scientific datasets in the INDIGO-datacloud project. In: CF 2015, pp. 343–348. ACM, New York (2017)
22. Yasin, A., Liu, L., Cao, Z., Wang, J., Liu, Y., Ling, T.S.: Big data services requirements analysis. In: Kamalrudin, M., Ahmad, S., Ikram, N. (eds.) APRES 2017. CCIS, vol. 809, pp. 3–14. Springer, Singapore (2018). https://doi.org/10.1007/978-981-10-7796-8_1
23. Fernandez-Garcia, A.J., Iribarne, L., Corral, A., Wang, James Z.: Evolving mashup interfaces using a distributed machine learning and model transformation methodology. In: Ciuciu, I., et al. (eds.) OTM 2015. LNCS, vol. 9416, pp. 401–410. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26138-6_43
24. Ji, J., Peng, R.: An analysis pattern driven requirements modeling method. In: REW Workshops, IEEE International, pp. 316–319. IEEE Press (2016)
25. Santos, J.C., et al.: BUDGET: a tool for supporting software architecture traceability research. In: WICSA 2016, pp. 303–306. IEEE Press (2016)
26. Nesi, P., Pantaleo, G., Sanesi, G.: A hadoop based platform for natural language processing of web pages and documents. J. Vis. Lang. Comput. **31**, 130–138 (2015)
27. Zhang, Y., Chen, Y., Ma, Y.: A framework for data-driven automata design. In: Liu, L., Aoyama, M. (eds.) Requirements Engineering in the Big Data Era. CCIS, vol. 558, pp. 33–47. Springer, Heidelberg (2015). https://doi.org/10.1007/978-3-662-48634-4_3

Pattern Library for Use-Case-Based Application Logic Reuse

Michał Śmiałek(✉), Albert Ambroziewicz, and Rafał Parol

Warsaw University of Technology, Warsaw, Poland
smialek@iem.pw.edu.pl

Abstract. This paper discusses the concept of patterns at a relatively early phase in software lifecycle, where detailed user-system dialogue (application logic) is defined. The dialogue is captured in generalised sequences of interactions performed by the system and its users, precisely linked with abstract domain vocabulary elements. We group individual interactions into sets of short scenarios which constitute "snippets" of system's observable behaviour. In the paper we present several example patterns that form an initial library. We substantiate validity of the library with an example instantiation of patterns into a full and detailed use case specification. This instantiation consists in selecting patterns, combining them together and substituting abstract vocabulary elements with concrete ones. The resulting concrete application logic models can then be used as input to further automatic processing, including application code generation.

Keywords: Use cases · Patterns · Application logic · Reuse

1 Introduction

Contemporary software systems present high repeatability in their structure and logic (behaviour). It is an obvious desire of software developers to be able to reuse the reoccurring elements within the various artefacts they produce during the software lifecycle, following the idea Alexander presented in his work on the "pattern language" [3]. Even though most software systems at least hint at some repeatability in their structures, we claim that no satisfactory solution to embrace such patterns exist at the level of functional requirements (see e.g. insights from Naish and Zhao [15]).

In this paper we concentrate on repeatability of application logic as expressed within descriptions of use cases. Use cases, as introduced originally by Jacobson [9] constitute means to specify goal-driven sequences of user-system interactions that express the logic of the given application. Nowadays, use cases became one of the dominating approaches to specify functional requirements. The typical way to express the use case contents is through textual or graphical (e.g. activity diagram) scenarios. There exist various templates and styles for writing use case scenarios (see e.g. [2,5,11]). Some of the approaches propose to define certain

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 90–105, 2018.
https://doi.org/10.1007/978-3-319-97571-9_9

patterns in system behaviour expressed through use case models and their scenarios. At the most basic level, this involves capturing reoccurring arrangements of use cases [19] ("relationships between the ovals"). However, the most interesting from our point of view are the patterns that involve the internals of use cases - "contents of the ovals" [12].

So far, several approaches to introduce use case content patterns were proposed. Some of these approaches concentrate on providing patterns for use case specifications expressed mostly in natural language [6]. However, the most relevant for this paper are approaches that introduce certain restricted languages to define use case scenarios. Some of them propose formal grammars, expressed e.g. through EBNF rules [1,7]. In other cases, the pattern language syntax is expressed through a metamodel [1,14]. The ultimate goal is to be able to automate various tasks associated with requirements engineering [8], for instance – partial generation of formally expressed scenarios from less formal specifications [18].

The pattern library and its validation, which is the main contribution of this paper, is based on our previous work [4] where we give metamodel-based foundations for our pattern concept. The language in which we express the application logic is called RSL [10,24]. Here we present how this language can be used to express application logic. We also present some additional language constructs that facilitate pattern definition and application. What distinguishes our approach from previous ones is that our patterns constitute small "snippets" of application logic that can be combined in various ways to develop full use-case-based specifications. Further on, these specifications can be used to generate application code in an automatic manner (see [22,24]).

2 Application Logic Language

RSL provides a controlled grammar, which is used mostly to express the use case contents, i.e. the application logic. The grammar, being formal enough to be processed automatically is also close to natural language as it is based on simple sentences. Its simplicity encourages formulating precise and unambiguous texts. Ambiguity is prevented in RSL through the introduction of a central "glossary", a domain specification model, that contains domain entity attributes and dynamic features, along with relationships among these entities. Here we will present just a small and simplified excerpt of RSL, relevant for our pattern library. First, we present the abstract grammar of the language expressed as a metamodel in the MOF language [16]. Then, we give a brief example of the language's concrete syntax and semantics.

Figure 1 presents part of the syntax to express application logic scenarios. The top level syntactic element is the UseCase which has similar semantics to that found in UML. Contents of a UseCase can be represented by several ConstrainedLanguageScenarios. Each scenario has one or more numbered ScenarioSentences as scenario steps. There are several types of sentences but here we mention just two of them, and concentrate on the SVOSentence. SVO(O)

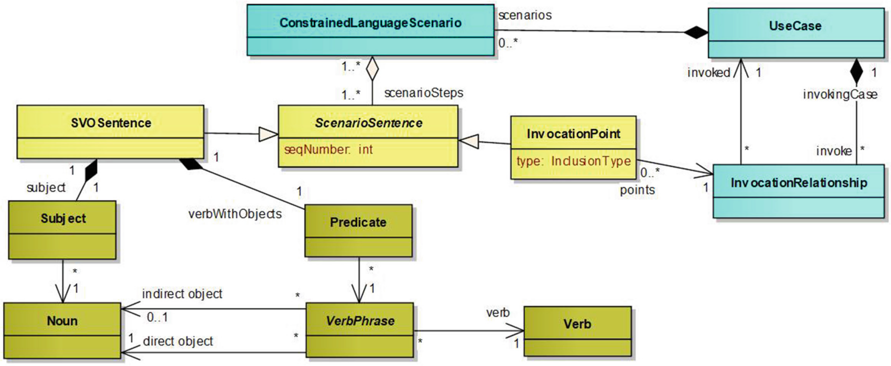


Fig. 1. Simplified metamodel for application logic scenarios

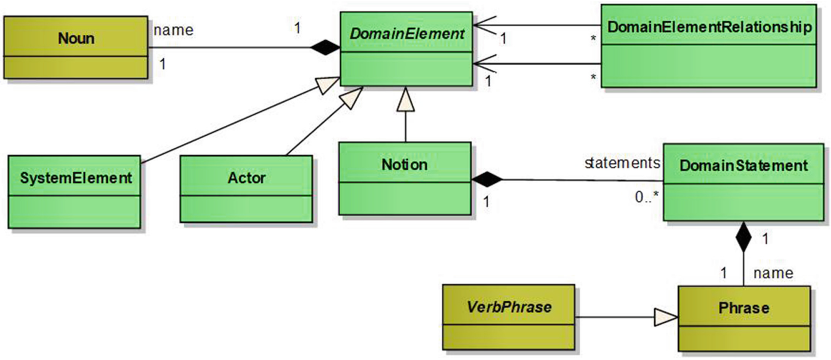
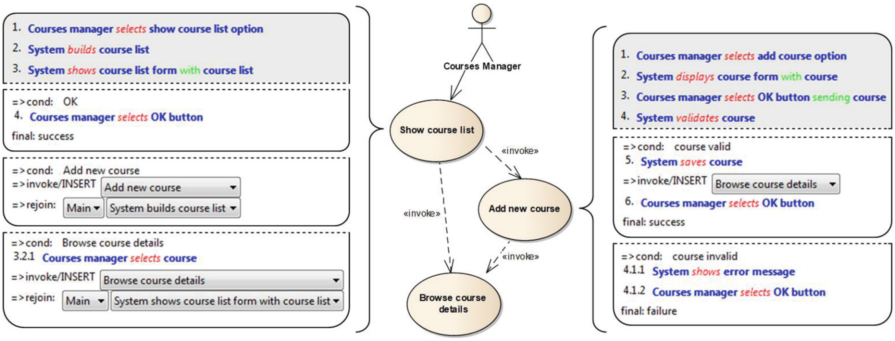


Fig. 2. Simplified metamodel for domain vocabularies

sentences are simple sentences consisting of a Subject and a Predicate group. The subject links to a single Noun. The predicate group links to a VerbPhrase which points at a Verb and two Nouns (a direct and indirect object). Phrases, Verbs and Nouns are parts of a centrally defined vocabulary, as presented in Fig. 2. Domain vocabularies consist of DomainElements with DomainElementRelationships. Every DomainElement has a name which is a Noun. Notions, which are special kinds of DomainElements contain DomainStatements that are composed of Phrases (including VerbPhrases).

This syntax allows us to unify and enunciate vague semantics of use cases (see discussions by Simons [20] and Śmiałek [21]). In addition, we introduce the InvocationRelationship that denotes invoking scenarios of a use case from within another use case. After performing one of the final actions in the invoked use case (steps of the scenario), the flow of control returns to the invoking use case right after the point of invocation (a specific position in the scenario) to perform the remaining part of the invoking use case. In scenarios, invocations are introduced through special sentences of type InvocationPoint.



1. Courses manager selects show course list option
2. System builds course list
3. System shows course list form with course list

=>cond: OK
4. Courses manager selects OK button
final: success

=>cond: Add new course
=>invoke/INSERT Add new course
=>rejoin: Main System builds course list

=>cond: Browse course details
3.2.1 Courses manager selects course
=>invoke/INSERT Browse course details
=>rejoin: Main System shows course list form with course list

1. Courses manager selects add course option
2. System displays course form with course
3. Courses manager selects OK button sending course
4. System validates course

=>cond: course valid
5. System saves course
=>invoke/INSERT Browse course details
6. Courses manager selects OK button
final: success

=>cond: course invalid
4.1.1 System shows error message
4.1.2 Courses manager selects OK button
final: failure

Fig. 3. Example use case model with scenarios

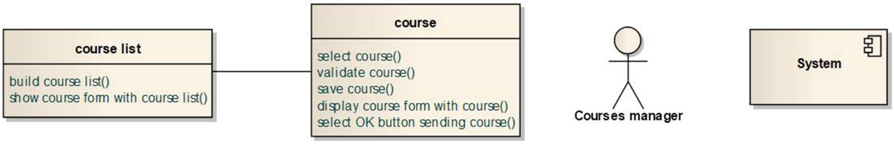


Fig. 4. Example domain vocabulary

Figures 3 and 4 show an example of concrete syntax of RSL. This includes also additional constructs like conditional sentences and final sentences, and invocation point sentences. Phrases used within SVO sentences are contained in the vocabulary shown in Fig. 4. More information on the RSL syntax and semantics can be found elsewhere [24].

To define the pattern library we introduce an extension which we call RSL-AL [4]. An excerpt from the abstract syntax is shown in Fig. 5, and an example of concrete syntax in Fig. 6. The additional elements include the InsertionRelationship and the associated InsertionPoint sentence. These elements will allow us to define small "snippets" of use case scenarios and link them (insert) into the courses of other scenarios. This linking can be done through matching InsertionPins and NotionPins. An InsertionPin points to an InsertionRelationship indicating the "snippet" UseCase to be inserted. The NotionPin points at a Notion that is to be matched between the inserting and inserted use case. Finally, we can also define LocalPreconditions which consist of a Verb and possibly a Modifier. Respective examples of these elements are given in Fig. 6.

3 Pattern Library

The above defined application logic language enables creating libraries of patterns. Here we propose an initial library of fundamental application logic building blocks. The structure of the library is shown in Fig. 7. The individual patterns and their relationships were found during the analysis of a large collection of

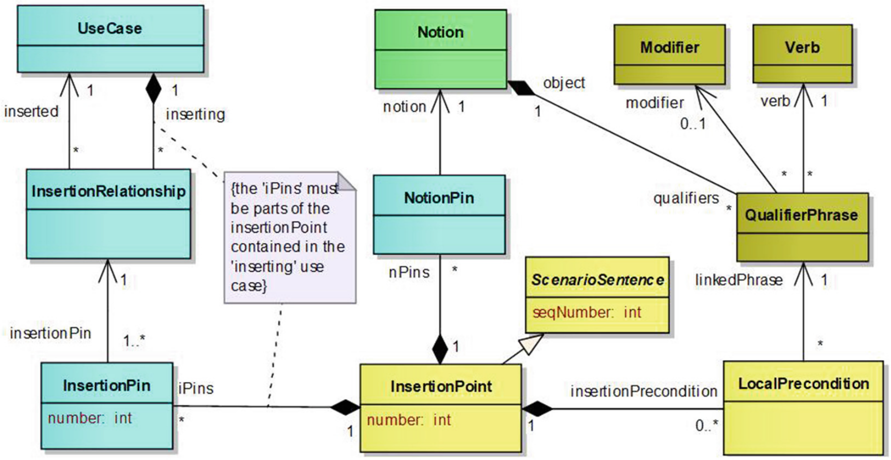


Fig. 5. Extended metamodel with pattern-specific elements

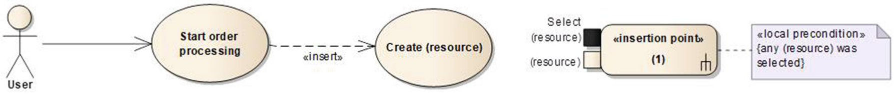


Fig. 6. Example notation for pattern-specific elements

so-called "software cases" (see [23]). These cases had their requirements specifications prepared using RSL with constructs similar to that of the pattern language presented above. There were analysed more than 50 software cases with more than 1000 use cases. The cases were prepared by the industrial and academic participants of the ReDSeeDS project (www.redseeds.eu) and students during classes on model-driven software development (see [26]). Our analysis of this vast material showed recurring logic despite differences between the individual problem domains for the various systems. The domains of the analysed systems ranged from fitness club and theme park, through web stores, fire brigade support, procurement, up to finance and banking. In the following, we describe some of the core application logic patterns. For each of the patterns we provide a short description presenting its potential usage, its logic (scenarios) in the concrete syntax of activity diagrams, and example applications.

1. **Create (resource).** *Usage:* This pattern contains interactions related to creating a new resource based on the actor input. New data is entered through a form an then validated (see "Validate (resource)"); in case of successful validation the resource is created. *Pattern logic:* see Fig. 8a. *Example applications:* registering personal details, entering a new shop item.
2. **Validate (resource).** *Usage:* This pattern contains interactions related to validating some previously entered data. The pattern does not depend on the data input method, only on the description of the resource for which the data

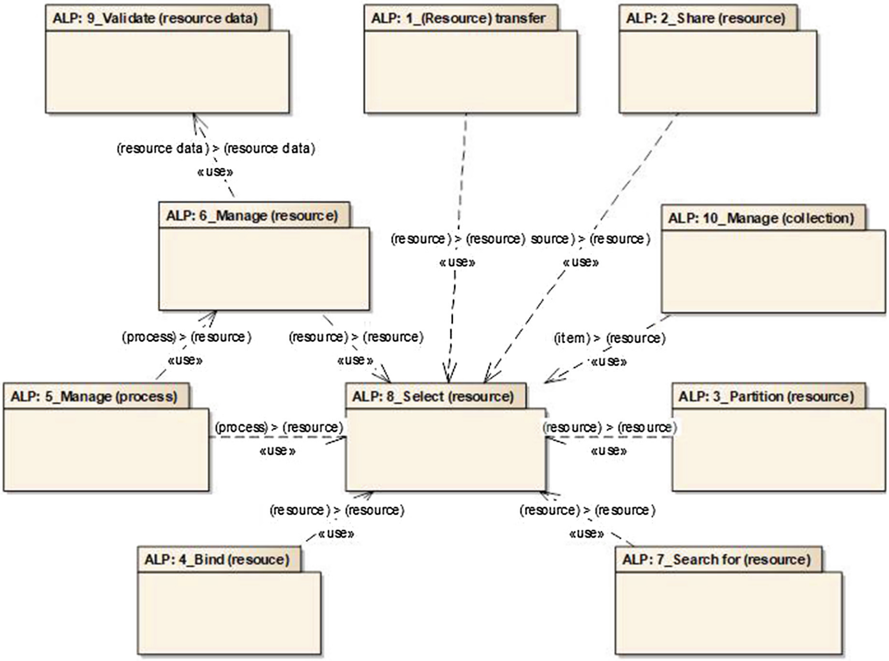


Fig. 7. Overview of the pattern library.

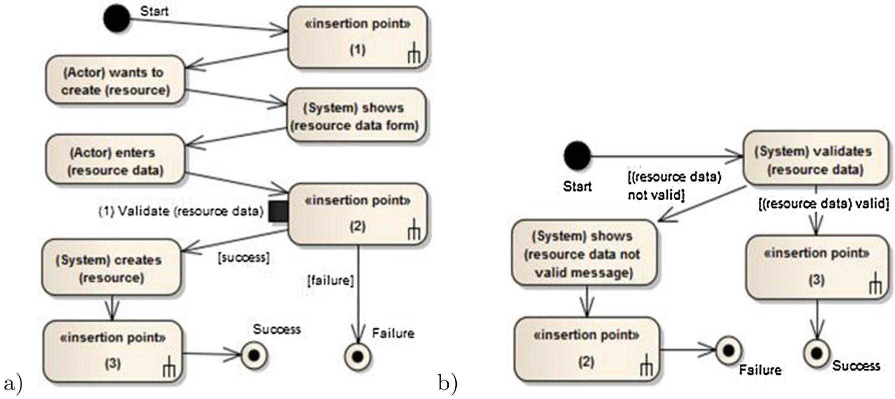

Start
«insertion point»
(1)
(Actor) wants to create (resource)
(System) shows (resource data form)
(Actor) enters (resource data)
«insertion point»
(2)
(1) Validate (resource data)
(System) creates (resource)
[success]
[failure]
«insertion point»
(3)
Success
Failure
a)
b)
Start
(System) validates (resource data)
[(resource data) not valid]
[(resource data) valid]
(System) shows (resource data not valid message)
«insertion point»
(3)
«insertion point»
(2)
Failure
Success

Fig. 8. Logic of the "Create (resource)" (a) and "Validate (resource)" (b) patterns

is validated. When validation fails, an appropriate message is shown. *Pattern logic:* see Fig. 8b. *Example applications:* validate personal details entered in a registration form, validate login information.

3. **Select (resource).** *Usage:* This pattern allows the system user to point at a specific resource of a given type by selecting it from a set of resources. This

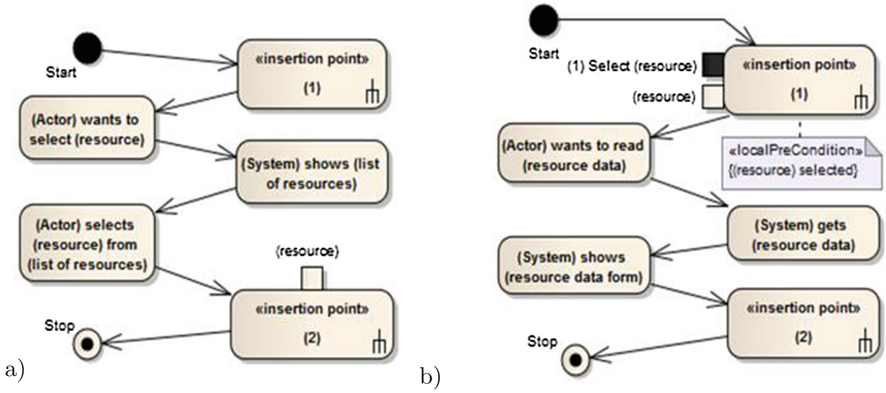

Start
«insertion point»
(1)
(Actor) wants to select (resource)
(System) shows (list of resources)
(Actor) selects (resource) from (list of resources)
(resource)
«insertion point»
(2)
Stop
a)
Start
(1) Select (resource)
(resource)
«insertion point»
(1)
(Actor) wants to read (resource data)
«localPreCondition»
{(resource) selected}
(System) gets (resource data)
(System) shows (resource data form)
«insertion point»
(2)
Stop
b)

Fig. 9. Logic of the “Select (resource)” (a) and “Read (resource)” (b) patterns

simple pattern is the basis for many other patterns, as the majority of them necessitate resource pre-selection. *Pattern logic:* see Fig. 9a. *Example applications:* selecting a running task from a list of operating system processes, selecting a recipient of an e-mail from a contact list.

4. **Read (resource).** *Usage:* This pattern allows to display data of some resource in a window form after first selecting it (see “Select (resource)”). The selected resource is retrieved and shown to the actor. *Pattern logic:* see Fig. 9b. *Example applications:* displaying personal details of a selected person, showing details of a selected shop item.
5. **Update (resource).** *Usage:* This pattern contains interactions related to editing a resource by an actor. The old resource data is presented in a window form and available for updating. After entering updated data, it is validated (see “Validate (resource)”); in case of successful validation the resource is updated. *Pattern logic:* see Fig. 10a. *Example applications:* updating personal details, changing data of a existing shop item.
6. **Delete (resource).** *Usage:* The logic of this pattern represents interactions leading to deletion of a given resource. In the first interaction step, the resource is selected by the actor. Then the actor selects to delete the resource which is deleted if confirmed. *Pattern logic:* see Fig. 10b. *Example applications:* removing personal details of a selected patient, deleting an existing shop item.
7. **Find (resource).** This pattern defines a typical retrieval process based on parameterised search criteria. An actor is presented a search criteria form, and the system looks for domain resources that match the criteria. Eventually a list of found elements is shown (see “Select (resource)”). *Pattern logic:* see Fig. 11a. *Example applications:* searching for a book in an on-line library, searching for files in a file system.
8. **Bind (resource).** *Usage:* The logic of this pattern represents abstract functionality for binding of two resources (assigning one resource to another).

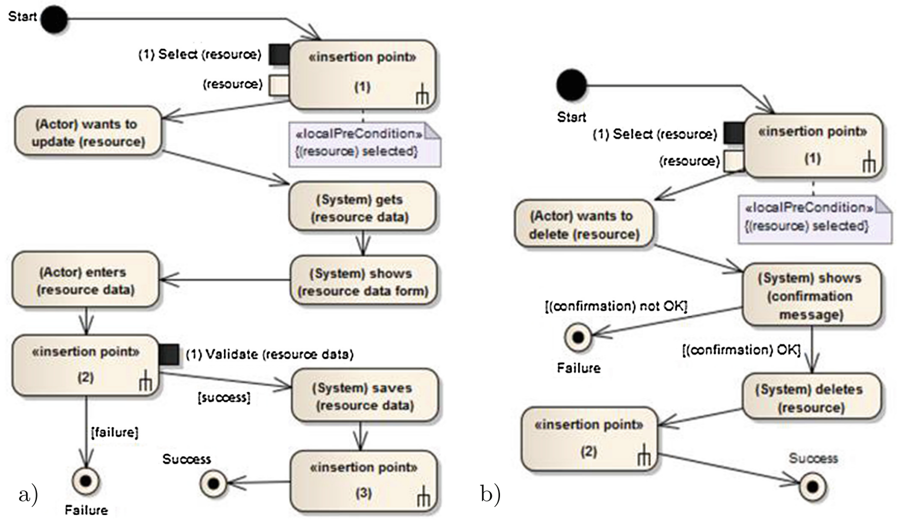


Fig. 10. Logic of the "Update (resource)" (a) and "Delete (resource)" (b) patterns

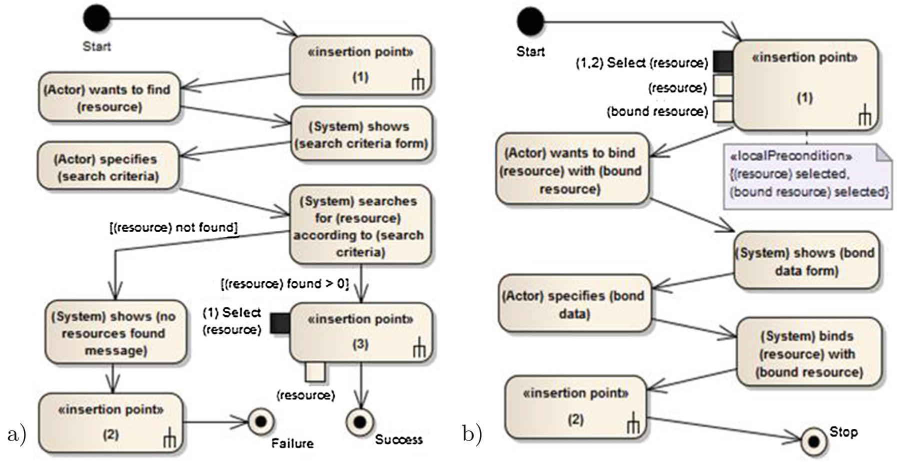

Start
«insertion point»
(1)
(Actor) wants to find (resource)
(System) shows (search criteria form)
(Actor) specifies (search criteria)
(System) searches for (resource) according to (search criteria)
[(resource) not found]
[(resource) found > 0]
(System) shows (no resources found message)
(1) Select (resource)
«insertion point»
(3)
(resource)
«insertion point»
(2)
a)
Failure
Success
Start
(1,2) Select (resource)
(resource)
(bound resource)
«insertion point»
(1)
(Actor) wants to bind (resource) with (bound resource)
«localPrecondition»
{(resource) selected,
(bound resource) selected}
(System) shows (bond data form)
(Actor) specifies (bond data)
(System) binds (resource) with (bound resource)
«insertion point»
(2)
Stop
b)

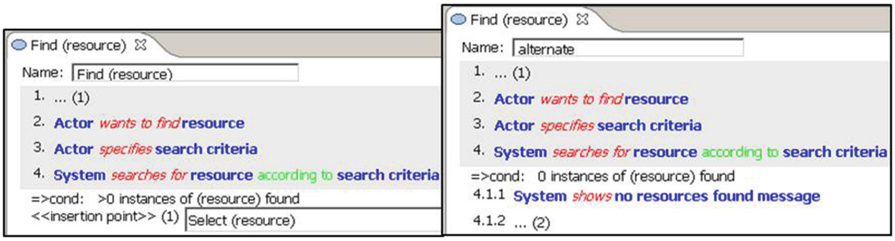
Fig. 11. Logic of the "Find (resource)" (a) and "Bind (resource)" (b) patterns

In the first interaction step both resources are selected by the actor, and then the system binds them in the manner specified by the actor. *Pattern logic:* see Fig. 11b. *Example applications:* assigning a doctor to a patient, attaching a testimony to a given court case.

As it can be noted, some of the presented patterns define application logic for typical data handling operations (cf. the well established CRUD pattern [13]). Of course, the above example patterns do not exhaust all the possibilities to generalise application logic. The library can be easily extended with additional “snippets” including those not listed in Fig. 7. On the other hand, we can observe that the presented patterns are very simple, and sometimes contain just a few sentences. Still though, we should notice that knowledge about the pattern layouts can facilitate and speed-up development of use case specifications. Also, we can create libraries that contain more complex logic and are oriented towards more specific applications. In the following section we will demonstrate how the pattern instantiation process can be organised.

4 Applying Patterns from the Library

In order to validate the presented library we have developed a plug-in to the existing ReDSeeDS tool [25] (see also www.redseeds.eu). The tool offers an editor for RSL and several tools for automatic processing of RSL models (including code generation). Our plug-in extends ReDSeeDS with a simple repository for storing the patterns and a tool to instantiate them. An example pattern defined in the tool is presented in Fig. 12. It shows the “Find (resource)” pattern from Fig. 11 in textual form. Instead of an activity diagram with two alternative paths, we define two scenarios in purely textual notation.



Find (resource)

Name: Find (resource)

1. ... (1)
2. Actor wants to find resource
3. Actor specifies search criteria
4. System searches for resource according to search criteria

=>cond: >0 instances of (resource) found

<<insertion point>> (1) Select (resource)

Find (resource)

Name: alternate

1. ... (1)
2. Actor wants to find resource
3. Actor specifies search criteria
4. System searches for resource according to search criteria

=>cond: 0 instances of (resource) found

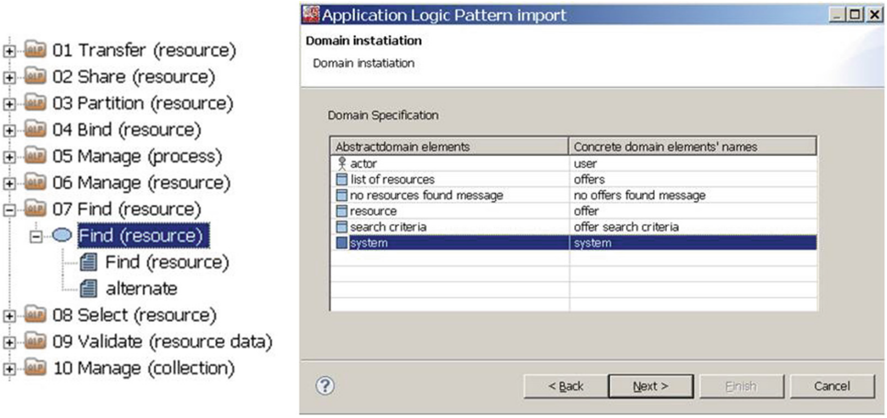
4.1.1 System shows no resources found message

4.1.2 ... (2)

Fig. 12. Example pattern in a tool

The process of instantiating a pattern is shown in Fig. 13. The developer selects a pattern and then the tool offers an instantiation dialogue window. In the window, the developer substitutes abstract domain elements found in the patterns (like “resource” or “search criteria”) with concrete ones. What is important, the concrete domain elements are taken from the existing domain model and can be reused for various pattern instances. The process is very quick and much shorter than typical copy-paste approach. Moreover, it significantly reduces the possibility of future inconsistencies in the requirements specification. The tool assures that all the scenario sentences are properly and unambiguously attached to the domain vocabulary elements. As a result, the developer receives a ready instance of the pattern that can be further modified or appended with other patterns at respective insertion points. Figure 14 presents an example instance of the pattern from Fig. 13.

With the above procedure we can quickly build a complete functional requirements specification from ready “snippets” taken from the library. However, the


- 01 Transfer (resource)
- 02 Share (resource)
- 03 Partition (resource)
- 04 Bind (resource)
- 05 Manage (process)
- 06 Manage (resource)
- 07 Find (resource)
  - Find (resource)
    - Find (resource)
    - alternate
- 08 Select (resource)
- 09 Validate (resource data)
- 10 Manage (collection)

Application Logic Pattern import

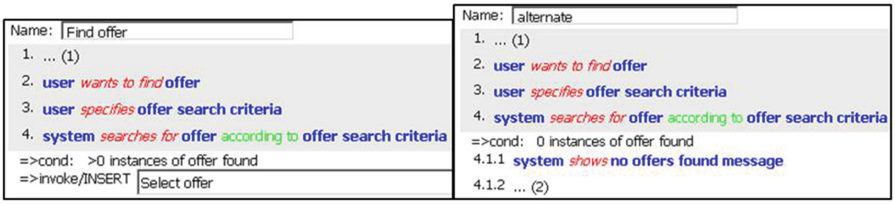
Domain instatiation

Domain instatiation

Domain Specification

Abstractdomain elements	Concrete domain elements' names
actor	user
list of resources	offers
no resources found message	no offers found message
resource	offer
search criteria	offer search criteria
system	system

Fig. 13. Creating an instance of a pattern


Name: Find offer

1. ... (1)
2. user wants to find offer
3. user specifies offer search criteria
4. system searches for offer according to offer search criteria

=>cond: >0 instances of offer found
=>invoke/INSERT Select offer

Name: alternate

1. ... (1)
2. user wants to find offer
3. user specifies offer search criteria
4. system searches for offer according to offer search criteria

=>cond: 0 instances of offer found
4.1.1 system shows no offers found message
4.1.2 ... (2)

Fig. 14. Example pattern after instantiation

patterns can be also applied if an appropriate automation tool is not available. The patterns can be stored in a word processor and applied manually through copy-paste. We can also use a CASE tool (e.g. a UML editor) to store the pattern use cases and scenarios as description text. In such a case, the pattern is applied by copying the given abstract use case from the library folder to the current project space. These two variants are more error-prone and necessitate more effort, as the instantiation needs to involve manual substitution of abstract notions to concrete ones.

To validate that our pattern library can be treated as a practical and convenient approach, even without a dedicated tool support, we will present a simple case study example. Our aim is to show how patterns can be used to build a complete use case specification when formulating requirements for a concrete software system. A simple web store application will play the role of our example here. As we can see in Fig. 15, our specification consists of 11 use cases (software requirements units), defined with the help of application logic patterns, inserted into each other according to the precise rules formulated and given in the previous sections. The size of the example has been adjusted to show, that even when facing some non-trivial requirements, we are able to cover majority of possible cases. We can describe the logic of most of the functional require-

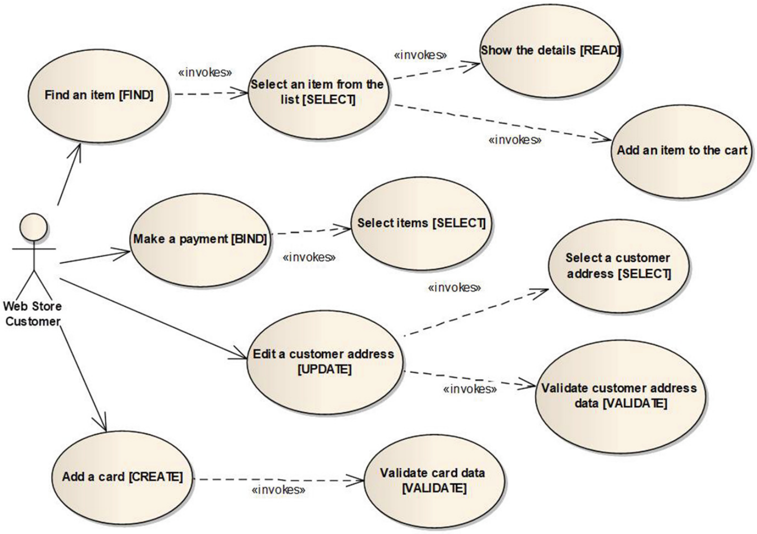
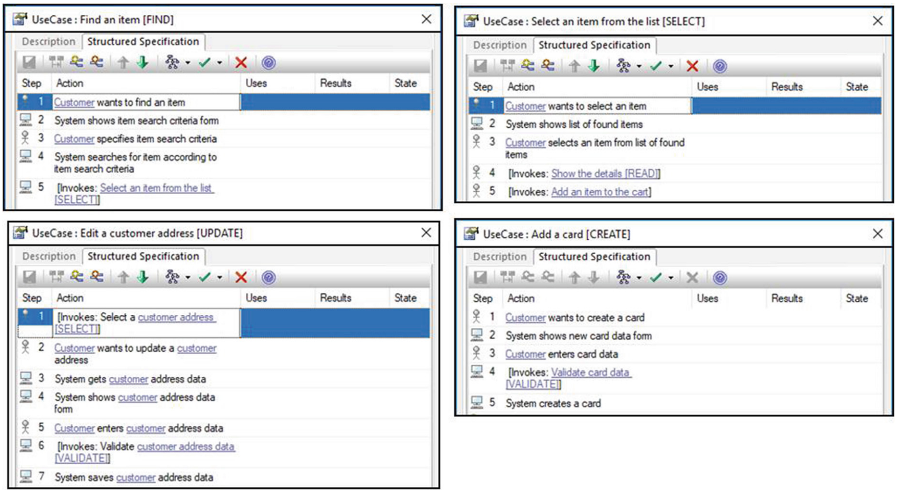


Fig. 15. Use case model built through applying patters

ments constituting the final specification, with just the use of application logic patterns. At the same time, limited amount of space in this paper allows us to present selected parts of the specification.

To specify the system, we start from the vision statement in natural language. The example system should allow for basic interaction between a customer and a web store application. The customer can look for a shopping item, examine detailed information that is available on this kind of item and add the item to the cart if interested in purchasing it. The ordered items need to be delivered to the customer's address and thus the customers needs to be able to update this kind of information, including selection of already defined addresses to be used for any future orders. Regarding payments, the system should allow to define (add) payment cards. These cards can be later used to make payments through assigning (cf. binding) them to specific orders.

Based on this initial description we can build the use case model through selecting appropriate patterns. We can observe that the abstract (resource) elements found in the various patterns should be now substituted by concrete notions from our current domain - "items", "cards" etc. Let's first take a look at the use case called *Find an item* (see top-left of Fig. 16). We can make use of the pattern called Find (resource), when trying to describe its logic. So, following the activity diagram presented in Fig. 11 and just by substituting the phrases in brackets, we obtain sentences like *Customer wants to find an item* instead of *(Actor) wants to find a (resource)*. In this particular scenario we decide to make use of the Insertion point no. 3 and insert an invocation to the "Select an item from the list" use case. This one is instantiated form the "Select (resource)" pattern and has one basic scenario as shown in Fig. 16 (top-right). The "Select



UseCase : Find an item [FIND]

Step	Action	Uses	Results	State
1	Customer wants to find an item			
2	System shows item search criteria form			
3	Customer specifies item search criteria			
4	System searches for item according to item search criteria			
5	[Invokes: Select an item from the list [SELECT]]			

UseCase : Select an item from the list [SELECT]

Step	Action	Uses	Results	State
1	Customer wants to select an item			
2	System shows list of found items			
3	Customer selects an item from list of found items			
4	[Invokes: Show the details [READ]]			
5	[Invokes: Add an item to the cart]			

UseCase : Edit a customer address [UPDATE]

Step	Action	Uses	Results	State
1	[Invokes: Select a customer address [SELECT]]			
2	Customer wants to update a customer address			
3	System gets customer address data			
4	System shows customer address data form			
5	Customer enters customer address data			
6	[Invokes: Validate customer address data [VALIDATE]]			
7	System saves customer address data			

UseCase : Add a card [CREATE]

Step	Action	Uses	Results	State
1	Customer wants to create a card			
2	System shows new card data form			
3	Customer enters card data			
4	[Invokes: Validate card data [VALIDATE]]			
5	System creates a card			

Fig. 16. Scenarios instantiated from abstract scenarios

(resource)" pattern has an insertion point at the end of its scenario with an associated notion pin. This indicated that we can insert additional invocations at this point. Here we have two of them ("Show the item details" and "Add the item to the cart"), both assuming that the "item" (cf. the (resource) pin) is selected. As we can see, "Add the item to the cart" is not related to any particular pattern and obviously we are free to introduce new use cases with more specific logic, not available for instantiation in the library. Finally, the two bottom scenarios in Fig. 16 illustrate instantiation of the same pattern (here: "Validate (resource)") for two different notions. The left scenario contains an invocation to "Validate customer address", while the right one - to "Validate card".

5 Discussion and Conclusion

The paper discusses ways of generalising application logic through proposing a library of elementary use-case-based patterns. The contents of this library consists of generic "snippets" of user-system interaction scenarios, intended for any problem domain. We should note that the presented patterns can be used as the basis for establishing specialised domain-specific libraries. The original library can be extended and reworked, depending on the requirements of a given problem and environment. Of course, a specialized library can be created from scratch using the proposed pattern language independently from the patterns proposed in this paper.

Our library is organized in a semi-formal way. The included patterns are described through models (activity and use case models) supplemented with textual descriptions. The contents of this library and any conformant library, allows the included patterns to be inter-related. These relations allow patterns to encapsulate other patterns as their components. This makes such libraries light-weight, both for their creators and their users: all repeatable elements of the library are abstracted and redundancy is limited. On the other hand, for the approach to be effective, the usage of such a library necessitates good awareness of its contents among its users. Still, we claim that analysis of other existing solutions shows that the presented approach has important practical benefits.

The presented library draws from experience gained through several larger case studies, that were part of the ReDSeeDS (www.redseeds.eu) and REMICS (www.remics.eu) projects. We have also applied the library during the analysis phase of a major governmental system in Poland. Observation of these case studies' results yielded some interesting findings. The "detachable" vocabulary concept used in the application logic language proves to be a very important advantage of the approach. This concept allows to separate concerns in problem descriptions. One part of a description is made in relation to the vocabulary (problem structure) and another part is made in regard to the behaviour (problem dynamics). From the point of view of the pattern user, the precisely linked vocabulary allows to reuse one aspect of the repeatable solution (behavior) and replace only the abstracted domain vocabulary. Also, this approach proved to be quite practical, as the pattern instantiation can be perceived as "smart copy-paste" operation. It makes the reuse process more efficient by reducing the number of errors committed and decreasing the time needed to apply the patterns. Other approaches are generally based on applying patterns through some kind of manual "copy-and-replace" mechanism, which in the source of frequent errors.

Another part of our approach that can be evaluated as important is the use of relationships between the behavioural elements at different levels of abstraction. In the application logic language, functionality descriptions can be managed at many levels: at the level of functionality groups, at the level of individual units of functionality, at the level of partial functionality contained in the units ("snippets"). The provided relationships allow to interrelate elements at each of these levels. The two available relationships (invocation and insertion) have proven to be practical. The insertion indicates existence of connections between the units of functionality (use cases and their "snippets"), which is important for factoring-out and distributing responsibility in any such pattern library. The invocation allows to preserve encapsulation of interaction flows and avoid the issue of use case interleaving.

Apart from the above qualitative observations we have also conducted several experiments. Though, due to space limitations and certain reservations regarding their validity here we will just present some insights. In one of such experiments (its early version was published in previous work [4]) the goal was to acknowledge levels of use case reuse using the presented approach and to determine gains in productivity when the tooling framework briefly presented in this work was used. The experiment was performed in two major steps. First, simple requirements specifications for customer-related parts of two systems in different business domains were created. The requirements models (use cases and scenarios) were created in RSL with the ReDSeeDS tool, but without using the pattern library deliberately. Thus, scenario sentences were written from scratch, using the RSL notation on these two specific problem domains. In the second step of the experiment, the patterns from the library were purposely used to create exactly the same application logic as in the first step.

During the experiment, quantitative analysis of analyst's performance was conducted. To measure the effort needed to create a given requirements model, a simple interaction step count was chosen (the intent was to measure the complexity of the user-system interaction description not the complexity of the system described by that interaction). The number of the reused elements was compared with the size of the original specification (first step) giving the resulting "reuse ratio". The experiment concluded that using the pattern library could decrease the effort needed for creating an exactly the same requirements specification without the library, by over 50%. However, this study had some threats to validity which mainly involved internal relations and timing of experiment steps. Despite this, our experience from other case studies and practical applications in industry work acknowledge these results. Still, we can see clear need to extend and improve experimental studies as part of future work.

Generally, our experience with using the library shows that the resulting requirements specifications can convey user requirements in a significantly clearer way than what can be found in typical products of requirements analysis. This means that such a specification contains a better description of the end-user needs and a more valuable material for developers and designers to work on during design and implementation. The patterns can be applied formally in a tool but also informally when sketching the application logic with the end-users (e.g. on paper). This clearly speeds-up the process, through taking and adapting parts of the specification from the pattern library.

It can be noted that proposed library uses certain concepts already available in UML [17] but in a refined way. For instance, the UML use case concept is redefined to some extent. The "include" and "extend" relationships are substituted with "invoke" and "insert". Moreover, the use case behaviour is made much more specific and precise through introducing scenarios and scenario sentences. This allows the resulting use case models to become the source for model-driven transformations and code generation [24]. Our future work in this respect will include refining the semantics of application logic language to become fully formal and precise. This will allow to treat this language as a kind of high-level programming language. We can notice that constrained language scenarios that use formally precise natural language can be used to "program" the upper (front-end) layers of a software system. This opens new interesting research avenues we intend to pursue in future work.

References

1. Aballay, L., Introini, S.C., Lund, M.I., Ormeno, E.: UCEFlow: a syntax proposed to structuring the event flow of use cases. In: 8th IEEE Computing Colombian Conference, pp. 1–6 (2013)
2. Adolph, S., Bramble, P., Cockburn, A., Pols, A.: Patterns for Effective Use Cases. Addison Wesley, Boston (2002)
3. Alexander, C., Ishikawa, S., Silverstein, M.: A Pattern Language: Towns, Buildings, Construction. Oxford University Press, New York (1977)
4. Ambroziewicz, A., Śmiałek, M.: Application logic patterns – reusable elements of user-system interaction. In: Petriu, D.C., Rouquette, N., Haugen, Ø. (eds.) MODELS 2010. LNCS, vol. 6394, pp. 241–255. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-16145-2_17
5. Cockburn, A.: Writing Effective Use Cases. Addison-Wesley, Boston (2000)
6. Díaz, I., Losavio, F., Matteo, A., Pastor, O.: A specification pattern for use cases. Inf. Manag. **41**(8), 961–975 (2004)
7. Georgiades, M.G., Andreou, A.S.: Patterns for use case context and content. In: Proceedings of the 13th International Conference on Software Reuse, pp. 267–282 (2013)
8. Issa, A.A., AlAli, A.I.: Automated requirements engineering: use case patterns-driven approach. IET Softw. **5**(3), 287–303 (2011)
9. Jacobson, I., Christerson, M., Jonsson, P., Övergaard, G.: Object-Oriented Software Engineering: A Use Case Driven Approach. Addison-Wesley, Reading (1992)
10. Kaindl, H., Smialek, M., Wagner, P., et al.: Requirements specification language definition. Deliverable D2.4.2, ReDSeeDS Project (2009). www.redseeds.eu
11. Kulak, D., Guiney, E.: Use Cases: Requirements in Context, 2nd edn. Addison Wesley, New York (2012)
12. Langlands, M.: Inside the oval: use case content patterns. Technical report, Planet Project (2014). v. 2
13. Martin, J.: Managing the Data Base Environment, p. 381. Prentice Hall PTR, Upper Saddle River (1983)
14. Misbhauddin, M., Alshayeb, M.: Extending the UML use case metamodel with behavioral information to facilitate model analysis and interchange. Softw. Syst. Model. **14**(2), 813–838 (2015)
15. Naish, J., Zhao, L.: Towards a generalised framework for classifying and retrieving requirements patterns. In: 1st International Workshop on Requirements Patterns, pp. 42–51 (2011)
16. Object Management Group: OMG Meta Object Facility (MOF) Core Specification, version 2.4.1, formal/2013-06-01 (2013)
17. Object Management Group: OMG Unified Modeling Language, version 2.5, ptc/2013-09-05 (2013)
18. Ochodek, M., Koronowski, K., Matysiak, A., Miklosik, P., Kopczynska, S.: Sketching use-case scenarios based on use-case goals and patterns. In: Proceedings of the XVIIIth KKIO Software Engineering Conference, pp. 17–30 (2017)
19. Overgaard, G., Palmkvist, K.: Use Cases: Patterns and Blueprints. Addison Wesley, Reading (2005)
20. Simons, A.J.H.: Use cases considered harmful. In: Proceedings of the 29th Conference on Technology of Object-Oriented Languages and Systems, pp. 194–203 (1999)

21. Śmiałek, M.: Accommodating informality with necessary precision in use case scenarios. J. Object Technol. **4**(6), 59–67 (2005)
22. Śmiałek, M., Jarzebowski, N., Nowakowski, W.: Translation of use case scenarios to Java code. Comput. Sci. **13**(4), 35–52 (2012)
23. Śmiałek, M., Kalnins, A., Kalnina, E., Ambroziewicz, A., Straszak, T., Wolter, K.: Comprehensive system for systematic case-driven software reuse. In: van Leeuwen, J., Muscholl, A., Peleg, D., Pokorný, J., Rumpe, B. (eds.) SOFSEM 2010. LNCS, vol. 5901, pp. 697–708. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-11266-9_58
24. Śmiałek, M., Nowakowski, W.: From Requirements to Java in a Snap: Model-Driven Requirements Engineering in Practice. Springer, Heidelberg (2015)
25. Smialek, M., Straszak, T.: Facilitating transition from requirements to code with the ReDSeeDS tool. In: 20th Requirements Engineering Conference, pp. 321–322 (2012)
26. Szmurło, R., Śmiałek, M.: Teaching software modeling in a simulated project environment. In: Kühne, T. (ed.) MODELS 2006. LNCS, vol. 4364, pp. 301–310. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-69489-2_37

Impact of Demographic Differences on Color Preferences in the Interface Design of e-Services in Latvia

Jurģis Šķilters[1](✉), Līga Zariņa[1], Signe Bāliņa[2], and Dace Baumgarte[2]

[1] Laboratory for Perceptual and Cognitive Systems at the Faculty of Computing, University of Latvia, Raina bulvaris 19, Riga LV-1050, Latvia
lpcs@lu.lv

[2] Faculty of Business, Management and Economics, University of Latvia, Aspazijas bulvaris 5, Riga LV-1050, Latvia

Abstract. In our study, we test users' color preferences of interfaces of typical electronic services' (e-services) environment. The motivation of our study is to explore discrepancy between (a) the high degree of availability of e-services in Latvia, and (b) low usage of these e-services among population. We aim to find the regularities regarding color preferences that could be useful in the development of e-services and that could support the rise of the usage of e-services in future. Although there are several reasons why people avoid using e-services (some of them socio-cognitive, some – technological), we would like to focus on a particular aspect – the colors of e-services interface. In particular, we test color preferences in respect to the interface depending on different demographic factors. Our hypothesis is that color contributes to the preference of using e-services and that such factors as gender, age, place of residence, education field, language knowledge, occupation, and hobbies determine the color preferences.

Keywords: Colors · Demographic factors · e-services · Interfaces

1 Introduction and Theoretical Framework

There is a discrepancy between the availability of electronic services and their relatively low use in population in Latvia. Currently more than 480 services are available digitally via the portal www.latvija.lv that is coordinated at the governmental level. This number of digitally accessible services is continuously increasing. However, despite the availability and flexibility of numerous public e-services provided by public, governmental, and private organizations, users frequently do not use those services. Preference to use services onsite – in a face-to-face communication – rather than use e-services is a very typical and frequent situation in Latvia.

Since year 2013 the percentage of fully available e-services has been continually growing in Latvia and also the usage of e-services has been growing from 35% (2013) to 69.5% (2016). However, as indicated in Table 1, activity of using e-services is lower when specific actions with documents are required, e.g., sending electronic documents

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 106–120, 2018.
https://doi.org/10.1007/978-3-319-97571-9_10

to a state institution. This opportunity has been used by 30.8% of inhabitants (2016), although documents signed with e-signature are generally accepted by state's institutions in Latvia.

Table 1. Availability and usage of e-Services in Latvia (2013 – 2016).

Year	Electronically available e-services, % of life events	Users' communication with state institutions (% of the total number of inhabitants)	Users sending filled-out forms electronically (% of the total number of inhabitants)
2013	73.1%	35.4%	12.6%
2014	81.6%	53.5%	19.0%
2015	85.4%	52.1%	29.1%
2016	n/a	69.5%	30.8%

Source: http://www.vraa.gov.lv/lv/

The mentioned discrepancy is the underlying problem that we are attempting to explore in our study in using two aims both sensitive to a variety of demographic variables: (a) to determine a subset of factors that impact the low usage of e-services, (b) to elaborate a reliable set of color preferences that can be used to generate a more user-friendly interface system. Although we assume that both socio-cognitive and perceptual factors determine preferences in using e-services, in this study we focus on perceptual determinants and, in particular, we explore color preferences and their impact on UX according to different demographic profiles of the users of e-services. Our findings are significant for improving the quality of information systems from the user perspective.

1.1 Color in UX Frameworks

Typically (but not always) UX analysis is limited by only few participants belonging to one or few demographic groups. UX-analysis frequently also lacks evidence from both cognitive and perceptual analysis' point of view. However, there are several well-discussed and widely accepted principles and default assumptions of color application in interface and UX-design [2]: (a) bright colors or blinking graphics have negative effects on productivity; (b) color should not convey the main information of the interface since there are people with deficits in color perception; therefore shape information should be primary in respect to color; a derived principle from this says that potential disabilities of the users have to be taken into account; (c) instead of bright colors pastels are recommended because in the longer use bright colors may induce visual fatigue, distraction and distaste; (d) color conventions might have impact on usability (red should be reserved for urgency contexts); (e) combination of red and blue should be avoided because it might have impact on focusing and cause visual fatigue and blurriness due to chromostereopsis (when substantial amounts of adjacent pure red and blue are in the single color combination; (f) blue should be in general avoided for text because pure blue is difficult for reading in terms of retinal processing; (g) allow controlling of presentational parameters (color, sound, text size etc.) according to user preferences (which might be important for disabled users); and, if possible and if the context is appropriate, enable color and general interface personalization; (h) colors should be used consistently with the message and across different periods of time in the interface system.

When an interface (in our case – an e-service) is used, the first impression can be induced by purely visual, aesthetic and emotional factors that can influence the further usage of that interface. In fact, a visually appealing (but hardly usable) website is preferred to a usable but not appealing one; usability matters but eventually at a later point (cp. [2]). Visual factors in a successful case of interface design should facilitate and promote the usage of the product (in our case – a particular e-service). Further, users are also diverse both in terms of age, education, occupation, and in terms of the aim of using certain e-service. We hypothesize that these factors impact the usage of e-services systems but to a different degree each. (For a recent study on the impacts of color on brand communication in social media, see [16]).

In our study, we test colors and their appropriateness in interfaces of typical e-services' environment in a more comprehensive sample covering diverse demographic profiles (consisting of variables such as age, place of residence, gender, occupation, field of education, knowledge of language, hobbies etc.). We explore the colors that are or are not preferred according to those different demographic profiles.

The analysis of the impact of demographic variables on color preferences have an added value for applied research: it is important for customer decision analysis in interface settings and therefore provides valuable recommendations for a focused and successful interface design.

1.2 Color Preferences

Color preferences seem to be highly variable according to general and specific object knowledge [9, 12], season [4], e.g., color differences between fall and other three seasons seem to have higher impact on color preferences [11] and culture [17]. Color preference is also linked to variability in emotions [6–8, 13] in a way that emotions evoked by colors can be distinguished and is at least to some extent determined also by the object knowledge associated to these colors [9]. Further, color preferences are age dependent, e.g., infants prefer more dark yellow, but less – light blue than adults [13], and color preferences differ also in childhood [14].

In this study, we focus on color preferences according to different demographic variables (cp. also [8]; for the impact of factors such and gender, age, and color on attentional processes in workplace cp. [10]) and we leave other factors such as color and emotion mappings for another study.

2 Method

2.1 Participants

In total 820 persons (school teachers (55%), librarians (33%) and municipality workers (10%)) from Latvia participated in the study. 91% of the sample were female and 9% male participants. The average age of the respondents was from 46–50, while 2% were less than 25 years old and 9% were over 61 years old. Most of the participants were native Latvian speakers (89%), with higher education (87%). Most of them have a degree in humanities or social sciences (68%), while 25% have an educational background in natural or engineering sciences. The sample was relatively balanced among Riga (the capital of Latvia (21%)), bigger cities of Latvia (48%), and country areas (31%).

The additionally obtained information showed that typical use of internet is between 2 and 5 h (64%) and is mainly related to work (90% of respondents indicate that they use internet for this purpose very frequently). Most of the participants (69%) use desktop computer as the primary tool (>60% of total use) for browsing for information (the second mostly used device is smartphone and the third – tablet). Most of the participants (46%) use 3–4 applications simultaneously; however, similarly large part (43%) uses just 1–2 applications simultaneously.

2.2 Stimuli and Setup

The survey's form consists of (a) question set referring to use of the e-services, (b) metacognitive questions, (c) questions concerning typical routines of using internet and digital devices, (d) questions related to color preferences, and (f) demographic information. For color preference testing a robust set of color stimuli was used (Table 2) in a way that no significant effects of display might impact the results. The colors that we included in our survey are conventional and simple, and are frequently used in interface default designs. We asked participants to rate their preference of each color's usage in an interface separately according Likert scale from 1 – 'like very much' to 5 – 'dislike very much'. The sequence of colors was randomized.

Table 2. Color stimuli used in the study.

Color HTML code	Red #ED1B24 ■	Orange #FF7F26 ■	Yellow #FEF200 ■	Green #23B14D ■	Blue #00A3E8 ■
	Violet #A349A3 ■	Pink #FEAEC9 ■	White #FFFFFF □	Black #000000 ■	

2.3 Task Design and Procedure

The survey template was generated with QuestionPro$^{\mathrm{TM}}$ tool. The survey was digitally distributed to the respondents as a link together with an instruction. The target audience of the survey covered teachers, librarians and municipality workers working with society and public education and it was reached through professional networks. The average time for completing the survey was 14–17 min.

After collecting the data, the data was summarized for the descriptive statistics. Hypothesis tests were performed regarding the differences of color preferences depending on demographic factors, routines of internet use, and the use of digital devices. The Kruskal-Wallis test was used due to the ordinal data type of the dependent variables and the significant differences ($\alpha = 0.05$) were detected by comparing the mean values in categories of each factor. Further, the correlation analysis (appropriate for ordinal data, i.e., Spearman, Kendall) was performed and significant correlations (0.01 level and 0.05 level) were obtained that, in turn, were used for detecting the general tendencies and modeling regressions (Ordinal regression). Additionally for the regression analysis the multicollinearity of independent variables was tested by evaluating Variance Inflation Factor (VIF). The assumption about proportional odds was tested with the Test of Parallel Lines. At the same time, factor analysis was performed to reduce the number of variables for regression analysis and to support interpretations of the obtained results. For extracting factors Principal Axis Factoring and Maximum Likehood methods were used and for rotation Oblimin and Varimax methods were used. Statistical processing was conducted by using IBM SPSS Statistics 22 software.

3 Results

3.1 Habits and Patterns in Device, Internet and e-Services' Use

According to our results demographic factors (as represented by our sample, cp. chapter 2.1) are linked with the typical routines of device use and, further, that both these aspects determine also the use of device's functionality (e.g., the choice for audio or video functions). The conducted statistical tests pointed to several significant differences ($\alpha = 0.05$) when comparing mean values (Fig. 1.) regarding factors determining patterns in device, internet and e-services' use.

As Fig. 1 shows, teachers tend to use less applications simultaneously, whereas municipality workers – more in average (for our sample structure cp. chapter 2.1). Teachers use less audio/video functions in desktop computers, but more than others – in mobile devices. In average, teachers also tend to use less internet daily than other two respondent groups in our sample. Accordingly, teachers are less open in using internet for e-services (e.g., formalities), entertainment, and communication; however, they use internet for educational activities comparatively more than others. Municipality workers tend to use smartphone for internet more frequently, while librarians use smartphones as well as tablets less than others. These tendencies correspond to those reflected in the use of audio/video functionality as well. Municipality workers and librarians use internet mostly for extended periods of time daily in average but they use

	AVERAGE	Teachers (55%)	Librarians (33%)	Municipality workers (10%)	Humanitarian (43%)	Social (25%)	Engineering (9%)	Natural (16%)	Male (9%)	Female (91%)	Riga (21%)	Cities (31%)	Country (48%)	Latvian (89%)	Russian (11%)	English language (77%)
Use - work *	4.87	4.85	4.94	**4.80**	4.87	4.87	4.92	**4.79**	**4.67**	4.89	4.84	4.85	4.91	4.87	4.88	4.88
Use - education *	3.80	**4.01**	**3.56**	**3.57**	3.85	3.81	3.85	3.93	3.76	3.80	3.75	3.84	3.75	3.78	3.91	**3.84**
Use - interests *	3.77	3.72	**3.81**	3.87	3.82	3.84	3.69	**3.61**	3.93	3.76	**3.92**	3.75	**3.70**	3.76	3.87	**3.81**
Use - purchases *	2.52	2.48	2.49	2.63	2.53	2.62	2.68	2.36	2.66	2.50	2.57	**2.60**	**2.34**	2.49	2.70	**2.60**
Use - formalities *	3.12	**3.03**	3.23	3.24	3.13	3.22	3.05	3.02	2.90	3.14	3.16	3.11	3.11	3.16	**2.86**	3.15
Use - entertainment *	3.14	**3.00**	**3.31**	3.26	3.16	**3.39**	3.15	**2.86**	3.31	3.12	**3.37**	3.13	**2.98**	3.14	3.06	**3.19**
Use - communication *	3.95	**3.84**	**4.09**	3.93	3.99	**4.09**	3.78	**3.71**	3.81	3.96	4.06	3.94	3.88	3.96	3.84	3.99
Audio/video - computer *	3.19	**3.08**	**3.42**	3.15	3.24	3.28	3.09	3.02	3.27	3.18	3.22	3.22	3.13	3.14	**3.59**	3.16
Audio/video - portative comp.*	3.06	**3.20**	**2.82**	3.22	3.10	3.16	3.19	3.10	3.07	3.06	3.11	3.08	3.00	3.08	2.88	3.11
Audio/video - tablet *	1.73	**1.79**	**1.61**	**2.00**	**1.63**	1.83	**1.95**	1.79	**2.10**	1.70	1.79	**1.84**	**1.51**	1.70	**2.02**	1.78
Audio/video - smartphone *	2.62	**2.63**	**2.48**	3.05	2.56	**2.93**	2.64	**2.46**	2.69	2.61	2.91	2.64	2.38	2.61	2.66	2.72
Internet use - h per day **	3.03	**2.73**	**3.42**	3.28	2.95	**3.25**	3.11	**2.80**	2.94	3.04	3.13	3.07	2.91	3.03	3.10	3.08
Internet use - desktop ***	8.75	**8.59**	**9.23**	**8.20**	8.95	**8.33**	8.78	8.73	9.10	8.71	**8.18**	8.66	**9.26**	8.84	**8.11**	**8.60**
Internet use - smartphone ***	3.23	3.34	**2.87**	**3.73**	3.09	**3.61**	3.20	3.21	2.99	3.25	**3.71**	3.29	**2.81**	3.18	3.63	**3.38**
Internet use - tablet ***	1.72	**1.76**	**1.62**	1.79	1.64	**1.69**	1.73	**1.80**	1.70	1.72	1.74	1.77	1.64	1.68	2.00	1.70
Application use ****	1.74	**1.69**	**1.73**	1.95	1.68	**1.84**	2.04	**1.83**	2.16	1.70	1.94	1.72	1.64	1.74	1.77	1.78
Informal/Formal *****	1.72	1.72	1.73	1.73	1.75	1.72	1.68	1.69	1.73	1.72	1.75	1.74	1.68	1.71	**1.84**	1.72

* Likert scale from 1 ='never' to 5 ='very often'

** 1=1h per day, 2=2-3hper day, 3=4-5h per day, 4=6-7h per day, 5=>7h per day

*** 1=0-10%, 2=10-20%, 3=20-30%, 4=30-40%, 5=40-50%, 6=50-60%, 7=60-70%, 8=70-80%, 9=80-90%

**** 1=1-2, 2=3-4, 3=4-5, 4=>6 applications simultaneously

***** Preference to 1=informal 'you' or 2=formal 'you'

Bold, underline – significant difference (α=0.05)

Fig. 1. Mean values characterizing patterns in device, internet and e-services' use depending on demographic factors

it comparatively less for education and work. In average, librarians use internet comparatively more for interests, entertainment and communication.

From the perspective of education, users with degree in engineering tend to use more applications simultaneously in average. In general, the field of education seems to be linked with preferences of the device format, and further in the choice of the audio/video functions. Participants with degree in humanities use less tablet for audio/video functions, while participants with degree in engineering – more than other groups. Smartphone for audio/video functions most frequently is used by respondents with degree in social sciences. That confirms also the general tendency of using internet in smartphones more frequently among respondents with education in social sciences. In contrast, the desktop computer is less frequently used in this sub-group. Respondents with social sciences' education tend to use internet mostly for extended periods of time daily whereas those with education in natural sciences use internet to a lesser extent daily. The tendencies of internet use also indicate that persons with education in social sciences use internet more for entertainment and communication while those with education in natural sciences use internet less than average for work, interests, entertainment, and communication.

Exploring the gender differences, the data shows that male participants use significantly more simultaneous applications than females. Male participants use tablets comparatively more for audio/video options, but in general they use internet to a lesser extent for work than females.

In respect to age differences, younger users prefer smartphone for internet navigation, but lesser – a desktop computer, whereas older users prefer desktop computer and accordingly lesser – smartphone (Fig. 2a). These differences are coherent with the use of audio/video function in smartphones. However, audio-visual functionality in portative computer is more frequently used by younger people and lesser by older participants (Fig. 2b). Younger participants seem to use internet more typically for education, purchases, and entertainment, and less frequently for work. This is contrary to the usage routines of older generation users (in particular, older than 46). These differences, however, are not continuous and show interruptions.

	Age group	21-25	26-30	31-35	36-40	41-45	46-50	51-55	56-60	>61
	% of respondents	2%	5%	8%	9%	14%	20%	17%	15%	9%
a	AVERAGE									
Audio/video - computer *	3.19	3.00	2.90	3.15	3.15	3.37	3.17	3.29	3.20	3.05
Audio/video - portative computer *	3.06	4.00	3.18	3.32	3.21	3.37	3.10	2.92	2.82	2.53
Audio/video - tablet *	1.73	1.63	1.98	1.97	2.07	1.98	1.69	1.58	1.55	1.40
Audio/video - smartphone *	2.62	4.00	3.43	3.47	3.16	2.96	2.53	2.24	1.89	2.12
b										
Internet use - desktop ***	8.75	7.38	6.45	7.47	7.47	8.25	8.95	9.57	9.80	9.88
Internet use - smartphone ***	3.23	4.75	5.25	4.48	4.31	3.71	2.91	2.55	2.28	2.39
Internet use - tablet ***	1.72	1.44	1.78	1.45	1.89	1.73	1.83	1.70	1.77	1.44

* Likert scale from 1 ='never' to 5 ='very often'

*** 1=0-10%, 2=10-20%, 3=20-30%, 4=30-40%, 5=40-50%, 6=50-60%, 7=60-70%, 8=70-80%, 9=80-90%, 10=90-100%

Fig. 2. Patterns in (a) use of audio/video functionality and (b) internet use depending on age

In terms of the place of residence, users living in Riga tend to use more applications simultaneously and more frequently smartphones and less frequently desktop computers which might be explained by the dynamics and flexibility of infrastructure and environment. In country, on the contrary, more frequently desktop computer is used and less frequently smartphones (which could be partly explained by everyday routines and constraints in high-speed mobile internet coverage). Participants in Riga use internet more for interests and entertainment whereas in country internet is used less frequently for purchases, which is more frequent type of internet use in other cities.

Regarding respondents that are not native Latvian speakers: a stronger preference of using polite, formal type of communication in internet can be observed in our results. In this group of respondent's internet is less used for e-services (e.g., variety of financial, residential and municipal formalities). That could be explained with the fact that the e-services provided by the state are in Latvian language.

Respondents with English language (as non-native) knowledge are more active consumers of technologies (in comparison with respondents without English knowledge) – they use more applications simultaneously and use internet in more extended periods of time daily; further, they use smartphone for internet more frequently and also more frequently use audio-video functionality. This group of users apply internet more frequently for such areas as education, interests, purchases, and entertainment.

In the light of hobbies, the data are summarized in Table 3, where the average values are represented for each dependent variable and the marked factors (hobbies) indicate statistically significant differences ($\alpha = 0.05$) depending on having certain hobby or not. For example, those participants that share the following hobbies – travel, using computer as a hobby tool, watching movies – use more applications simultaneously; in contrast, participants whose hobbies are reading or working in garden tend to use fewer applications simultaneously.

Analyzing the correlations between variables, some plausible relations can be observed. Even if most associations between variables are statistically weak, they still reflect some general tendencies. These tendencies of associations were also observed according to regression and factor analysis.

One of the tendencies that was discovered: the more applications are simultaneously used, the more audio-visual functionality is also used in all devices (r = from 0.140 to 0.276, 0.01 level). The use of audio-visual functionality seems to be linked with the intensity of use in different devices (e.g., the more frequently audio-visual functionality is used in tablet, the more it is also used in the smartphone (r = 0.224, 0.01 level)). At the same time the use of audio-visual functionality in different devices seems to correspond with the use of different devices for using internet (e.g., the more smartphone is used for internet, the more audio-visual functionality is used in mobile devices (portative computer (r = 0.189, 0.01 level), smartphone (r = 0.661, 0.01 level), tablet (r = 0.147, 0.01 level)) or on the opposite – the more desktop computer is used for internet, the less audio-visual functionality is used in mobile devices (smartphone (r = −0.562, 0.01 level), tablet (r = −0.370, 0.01 level). Finally, if smartphone is more used with audio-visual functionality, then other mobile devices are also more used with audio-visual functionality (portative computer (r = 0.330, 0.01 level), tablet (r = 0.224, 0.01 level)). Overall, it indicates that (a) there are respondents that are more technologically advanced and tend to use different devices for various functionalities, and (b) there are respondents that are more conservative in their habits of using technologies or they are basic technology users.

A plausible correlation indicating the activity pattern of multitasking individuals [5] was discovered: the more applications are used simultaneously the more hours daily internet is used (r = 0.262, 0.01 level). In general, the more applications are used (a) the more smartphones are used (r = 0.247, 0.01 level), and (b) there is a slight tendency that more frequently diverse aims in using internet are involved in the cases where multiple applications are used (these aims are, e.g., education (r = 0.111, 0.01 level), entertainment (r = 0.123, 0.01 level), communication (r = 0.146, 0.01 level), interests (r = 0.116, 0.01 level), purchases (r = 0.160, 0.01 level)). The increase in number of hours of internet use daily positively correlates with higher frequency of the use of audio-visual functionality in all devices (r = from 0.089, 0.05 level in tablet to 0.160, 0.01 level, in portative computer).

The survey's results shows an important consequence regarding multitasking behavior: the later in life (in terms of age) computer skills are acquired, the fewer applications are used simultaneously (r = −139, 0.01 level) and less audio-visual

Table 3. Differences in device, internet and e-services' use depending on common hobbies.

Hobby (% of respondents)	Use - work *	Use - education *	Use - interests *	Use - purchases *	Use - formalities *	Use - entertainment *	Use - communication *	Audio/video - computer *	Audio/video – portative computer *	Audio/video - tablet *	Audio/video - smartphone *	Internet use - h per day **	Internet use - desktop ***	Internet use - smartphone ***	Internet use - tablet ***	Application use ****	Informal/Formal *****
Average values	4.87	3.80	3.77	2.52	3.12	3.14	3.95	3.19	3.06	1.73	2.62	3.03	8.75	3.23	1.72	1.74	1.72
Reading (76%)	+					+				-					-	-	
Gardening (59%)				-		-				-	-		+	-		-	
Traveling (54%)				+					+		+		-	+		+	
Theatre (51%)							+										
Concerts (47%)							+										
Picking berries (43%)													+	-			
Computer (37%)			+	+		+	+		+	+	+	+	-	+		+	-
Education (35%)		+							+		+		-	+			
Culinary (33%)											+						
Social activities (25%)					+										-		
Sport (24%)											+			+			
Household (23%)										+							
Music (22%)			+			+	+	+	+		+			+			
Photo (22%)			+			+	+		+			+					
Dance (19%)								+									
Cinema (19%)			+	+		+	+	+	+	+	+	+	-	+		+	

* Likert scale from 1 ='never' to 5 ='very often'

** 1=1h per day, 2=2-3hper day, 3=4-5h per day, 4=6-7h per day, 5=>7h per day

*** 1=0-10%, 2=10-20%, 3=20-30%, 4=30-40%, 5=40-50%, 6=50-60%, 7=60-70%, 8=70-80%, 9=80-90%, 10=90-100%

**** 1=1-2, 2=3-4, 3=4-5, 4=>6 applications simultaneously

***** Preference to 1=informal 'you' or 2=formal 'you'

'- ' – negative tendency (significantly (α=0.05) lower average value than others in average)

'+' – positive tendency (significantly (α=0.05) higher average value than others in average)

functionality is used in mobile devices (laptops (r = −0.211, 0.01 level), smartphones (r = −0.379, 0.01 level), and tablets (−0.169, 0.01 level)). This could indicate that the period of life for acquisition of IT-skills might be a reasonable indicator of multitasking behavior. This would require a separate study, though. However, this is related to another general observation – the increase in age correlates negatively with increase in the number of applications used (r = −0.112, 0.01 level).

Part of the correlations confirms the results of the detected statistical differences of the means depending on demographic factors. Prospective multitasking behavior seems also to depend on the field of education, e.g., persons with background in engineering sciences tend to use more applications simultaneously (r = 0.105, 0.01 level). In terms of gender differences female participants use less applications simultaneously (r = −0.127, 0.01 level) which might be explained by the characteristics of our sample and should be interpreted with caution. However, other correlations of different strength can be observed without unambiguous explanation.

The regression models explaining variation in different device use were indicating that the main indicators are linked with age and different hobbies. But, for example, regarding the number of applications that are used simultaneously, the significant factors were gender, nationality, field of education, knowledge of language together with certain hobbies (reading, traveling, working in garden and computer as a hobby). Regarding the extent of daily internet usage, a significant factor was an occupation besides the factor of different hobbies (sport, computer). The aims of internet use were also linked with language knowledge and the field of education (besides hobbies, occupation, and age).

The factor analysis indicates the possibilities to distinguish several groups of respondents that can get characterized by some more general features, e.g., respondents that use tablets, respondents that use smartphones, respondents that use audio-visual functionality, respondents that use internet for activities that are not linked with work; and on the opposite – a group of respondents using internet for work and education, respondents that more use technologies, respondents whose hobbies are linked with culture or whose hobbies are characteristic for countryside.

3.2 Impact of Demographic Variables on the Color Preferences

Summarizing the survey's data about color preferences, green, blue, white, orange and purple are the colors that are mostly preferred as the interface colors in general (Fig. 3).

The analysis of data indicates significant differences ($\alpha = 0.05$) in color preferences within separated demographic factors. The summary is shown in the Table 4, where factors referring to the significant differences regarding mean values are marked.

There are differences in color preferences regarding the occupation of respondents. Teachers prefer purple more if compared to municipality workers who like it less.

Exploring the color preferences according to education, users with degree in humanities' background prefer more purple than the rest of the sample in average. Those users with background in social sciences rate red and yellow more negatively, whereas users with engineering background rate green, blue, purple, and pink more negatively. Examination of the levels of education shows that red is preferred in average more by those participants who do not have the higher education (this has to be interpreted with a caution because majority of participants (87%) in our sample has higher education).

Analysing the color preferences from the perspective of gender, we can observe that male participants prefer orange, yellow, blue, violet, and pink less than female. There is a tendency according to which male participants like less bright colors than female respondents.

The data shows unexpected results regarding the impact of the place of residence on the color preferences. This might be due to 'mere exposure effect' [1, 15] that white is less preferred in country whereas participants from cities prefer less bright colors.

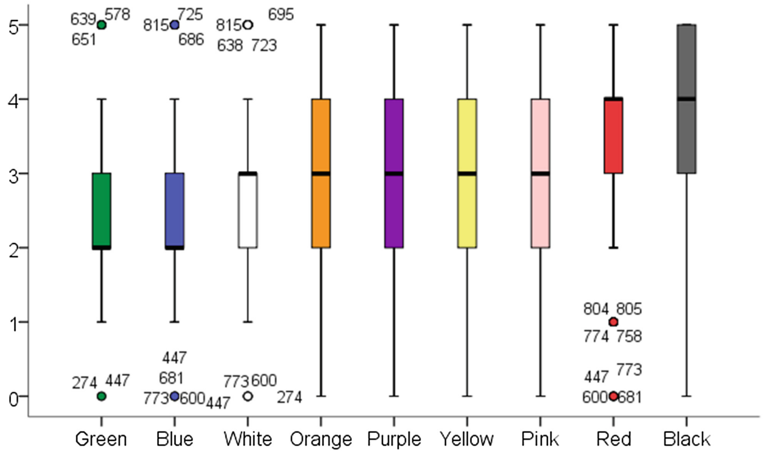


Fig. 3. The answers to the question: *How would you rate a given color in an interface?* according to the Likert scale ranging from 1 = ‘like very much’ to 5 = ‘dislike very much’ (Color figure online)

Table 4. Mean values characterizing regarding color preference in interface depending on demographic factors.

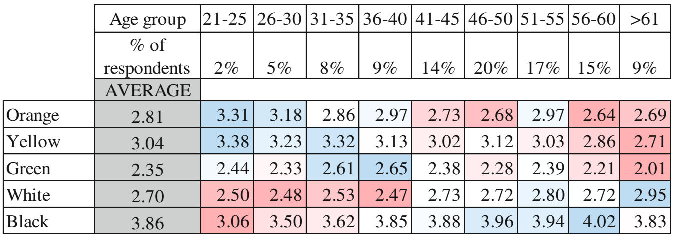
	AVERAGE	Teachers (55%)	Librarians (33%)	Municipality workers (10%)	Higher education (87%)	Humanitarian (43%)	Social (25%)	Engineering (9%)	Natural (16%)	Male (9%)	Female (91%)	Riga (21%)	Cities (31%)	Country (48%)	Latvian (89%)	Russian (11%)	English language (77%)
Green	2.35	2.31	2.32	2.48	2.37	2.27	2.44	**2.55**	2.46	2.47	2.33	**2.50**	2.36	**2.22**	2.37	2.17	**2.38**
Blue	2.54	2.49	2.60	2.59	2.55	2.52	2.53	**2.92**	2.54	**2.86**	**2.51**	2.60	2.51	2.55	2.55	2.40	2.54
White	2.70	2.67	2.69	2.66	2.68	2.74	2.71	2.76	2.61	2.66	2.70	**2.63**	2.64	**2.85**	2.69	2.68	**2.67**
Orange	2.81	2.77	2.77	2.90	2.83	2.84	2.89	2.92	2.84	**3.09**	**2.79**	**3.02**	2.75	**2.76**	2.82	2.73	2.85
Purple	2.86	**2.77**	2.86	**3.09**	2.87	**2.78**	2.91	**3.32**	2.83	**3.31**	**2.82**	2.95	2.84	2.84	2.85	2.88	**2.90**
Yellow	3.04	3.00	3.01	3.17	3.08	3.08	**3.20**	3.24	3.01	**3.34**	**3.01**	**3.21**	3.03	**2.94**	3.04	3.01	**3.10**
Pink	3.20	3.16	3.11	3.37	3.22	3.16	3.27	**3.54**	3.23	**3.54**	**3.16**	3.22	3.18	3.21	**3.24**	**2.84**	3.23
Red	3.58	3.59	3.56	3.48	**3.63**	3.65	**3.77**	3.72	3.59	3.71	3.57	**3.74**	3.57	**3.49**	3.60	3.43	**3.64**
Black	3.86	3.92	3.84	3.83	3.89	3.82	3.95	3.74	3.92	3.69	3.88	3.78	3.87	3.90	3.88	3.71	3.85

* Likert scale from 1=‘like very much’ to 5=’dislike very much’
Bold, underline – significant difference (α=0.05)

Examining in detail, the participants from Riga prefer more white and less red, orange, yellow, and green, while the relative opposite tendency is valid for the participants from country regions – they prefer more yellow, green, and red, and to a lesser degree white (if compared to other groups, i.e., bigger cities and Riga). Participants from bigger cities prefer more orange in comparison to participants from Riga and country areas, in average.

Significant differences occur when ethnic and linguistic factors are examined: in average, pink is more preferred among Russian population, whereas it is less preferred among Latvians. Also language knowledge shows some impact on color preferences: for example, participants lacking English knowledge prefer more red, yellow, green, and to a lesser degree white. However, these associations may be due to sample characteristics and perhaps are primary related to impact of the place of residence (the knowledge of English language is less common in country areas), while knowledge of other languages (e.g., Spanish or German) does not indicate any significant differences regarding color preferences as well. In terms of age differences, younger users like less orange, whereas older users (in particular, older than 45) tend to prefer orange (with exception of age group 51–55) (Table 5). The same pattern is valid in the case with yellow and green where these colors are less liked by younger users but preferred by older ones. An opposite tendency can be observed with black and white which is preferred by younger users and disliked by older ones.

Table 5. Differences in color preference in interface depending on age.



	Age group	21-25	26-30	31-35	36-40	41-45	46-50	51-55	56-60	>61
	% of respondents	2%	5%	8%	9%	14%	20%	17%	15%	9%
	AVERAGE									
Orange	2.81	3.31	3.18	2.86	2.97	2.73	2.68	2.97	2.64	2.69
Yellow	3.04	3.38	3.23	3.32	3.13	3.02	3.12	3.03	2.86	2.71
Green	2.35	2.44	2.33	2.61	2.65	2.38	2.28	2.39	2.21	2.01
White	2.70	2.50	2.48	2.53	2.47	2.73	2.72	2.80	2.72	2.95
Black	3.86	3.06	3.50	3.62	3.85	3.88	3.96	3.94	4.02	3.83

* Likert scale from 1='like very much' to 5='dislike very much'

Further, some hobbies also contribute to the color preferences (the results are summarized in Table 6. Impacts of hobbies on color preferences are hard to interpret unambiguously as they refer only to small sub-samples. For example, those participants involved in additional studies and learning prefer red to some lesser extent. According to the survey's results, orange is preferred by those users that indicate work in the garden as a hobby, but it is opposite for those participants who indicate sports or photo as their hobbies.

Analyzing the correlations between variables and the regressions, several patterns of a different associative or causal strength can be observed. These links were supported also by distinct factor groups after conducting factor analysis. These patterns provide us with tentative ideas for grouping characteristic tendencies together.

The higher is a preference for one color (except regarding black and white), the higher is also preference for other colors (r = from 0.131 to 0.373, 0.01 level). There are other interesting tendencies: the more white is preferred, the more black (r = 0.127, 0.01 level), pink (r = 0.110, 0.01 level), and yellow (r = 0.072, 0.05 level) are also preferred. And – the more black is preferred the more is also red (r = 0.245, 0.01 level), purple (0.116, 0.01 level), and white (0.127, 0.01 level) preferred. These two patterns

Table 6. Differences in color preferences depending on common hobbies.

Hobbies (% of respondents)	Green	Blue	White	Orange	Purple	Yellow	Pink	Red	Black
Average values	2.35	2.54	2.70	2.81	2.86	3.04	3.20	3.58	3.86
Gardening (59%)				+					
Traveling (54%)		+							
Concerts (47%)					+				
Picking berries (43%)			-			+			
Computer (37%)						-			+
Education (35%)								-	
Culinary (33%)		-							
Handicrafts (29%)							+		
Social activities (25%)	+								
Sports (24%)				-					
Music (22%)									+
Photo (22%)				-			-		
Cinema (19%)		+							+

Likert scale from 1='like very much' to 5='dislike very much'
'- ' – negative tendency (significantly (α=0.05) lower average value than others in average)
'+' – positive tendency (significantly (α=0.05) higher average value than others in average)

contribute to the explanation of preferences for dark or light colors and for a consistent contrast among colors in terms of pattern of preferences.

An interesting observation emerges when gender differences in color preferences were explored: females prefer more blue (0.099, 0.01 level), pink (0.109, 0.01 level), purple (0.137, 0.01 level) and there are positive tendencies also regarding orange (0.088, 0.05 level) and yellow (0.084, level 0.05). That is coherent with our results regarding differences in mean values (Table 3) where female respondents bright colors rate higher.

Examining the period of life when computer skills are acquired, we observed that participants, who acquired computer skills later in their life, indicate higher rates of preferences of yellow ($r = 0.141$, 0.01 level) and green ($r = 0.157$, 0.01 level) (and to a lesser degree – red, orange, pink, $r < 0.01$, level 0.05)), whereas black and white are less preferred ($r = -0.121$ and $r = -0.91$, 0.01 level). This pattern might also be linked to the age of participants (cp. tendencies by comparing the mean values (Table 4): the older participants the more they tend to prefer yellow ($r = 0.144$, 0.01 level), green (0.139, 0.01 level), and orange ($r = 0.087$, level 0.05); the older the participants the more negative correlations were observed regarding white ($r = -0.127$, 0.01 level) and black (−0.112, 0.01 level). Thus, our study also supports the hypothesis that age impacts color preferences (cp., e.g., [13]).

A less substantial and more inconclusive source of variability among color preferences is the field of education: subjects with degree in humanities have higher ratings of preferences of green ($r = 0.082$, level 0.05), whereas those with social sciences' background have lower ratings of preferences of red (r = −0.106, level 0.01) and yellow (r = −0.080, level 0.05). Participants with engineering background have lower ratings of blue (r = −0.121, 0.01 level), purple (r = −0.147, 0.01 level), pink (r = −0.105, level 0.01), and green (r = −0.081, level 0.05). This set of results, however, might also be explored and explained in the context with participants' occupation and other specific characteristics of our sample. Impact of hobbies on the color preferences seems to be less conclusive and harder to explain.

Although more studies of color preferences regarding the choice of specific devices (e.g., smartphones) are needed, the systematic differences that we have discovered in our results are convincing. Though our results are in some cases tentative, they can be plausibly used for recommendations in fields of interface design.

4 Discussion and Conclusions

Our results support the view of color preferences as dependent on several demographic variables. Our results are also important and unique because we were linking (a) typical routines of digital device use, (b) typical patterns of e-services' use, and (c) demographic variables (as the primary independent variables) with (d) color preferences. They also support (although from a different angle of analysis) other studies emphasizing the color variability and dependency on age [11, 13, 14], gender [4], cultural and ethnic differences [17]. According to our results, we would like to agree with the approaches arguing that categorical perception might have impact on color categorization [3].

In our study, we have discovered some tendencies that have not been explored in detail before, e.g., effects of educational background are among the impacts previously largely unexplored. Although a more careful experimental setting is needed to say more about these impacts, we can observe interesting (although yet tentative) results in color preferences (e.g., subjects with engineering background like less green, blue, purple, and pink). To answer, why these impacts exist and how do they fit together, separate studies (with a clear causal experimental setting) on each of the independent variables would be necessary but we might hypothesize that the frequency of interaction and some kind of mere exposure effects (e.g., [1, 15]) might shape the resulting color preferences.

According to our results, we might also emphasize that the preference of certain e-services is to some extent co-determined by preferences towards certain colors. Our core argument is that demographic variables have impact on color preferences and typical internet use; further we assume that these demographic variables significantly shape the way certain interface color is preferred. To our knowledge, this is the first study at this moment providing empirical evidence for that. The results of our work can be used for personalization of user interfaces according to users' demographic profiles and, therefore, can be implemented to substantially improve the user-friendliness of the e-services in terms of their colors.

Our study has certain limitations: (a) we do have mainly female participants and (b) representing 3 different but closely related areas of occupation (teachers, librarians, and municipality workers) which might also be seen as an advantage since these areas are among the most frequent users of e-services. Further, (c) more careful experimental analysis of each variable is needed in order to provide a more conclusive answer about the nature of the impact, e.g., we do not have conclusions about impacts of object perception on color preferences [9] and about the systematic way emotions shape color categorization [6–8]. Finally, due to the specifics of our survey we have not covered seasonal impacts on the variability of color preferences [4, 11].

References

1. Grimes, A., Kitchen, P.J.: Researching mere exposure effects to advertising-theoretical foundations and methodological implications. Int. J. Res. Mark. **49**(2), 191–219 (2007)
2. Hartson, R., Pyla, P.S.: The UX Book: Process and Guidelines for Ensuring a Quality User Experience. Elsevier, Amsterdam (2012)
3. Holmes, K.J., Regier, T.: Categorical perception beyond the basic level: The case of warm and cool colors. Cogn. Sci. **41**(4), 1135–1147 (2017)
4. Nelson, R., Schloss, K.B., Parker, L., Palmer, S.E.: Color preference: seasonal and gender differences. J. Vision, **12**, 72 (2012). https://doi.org/10.1167/12.9.72
5. Ophir, E., Nass, C., Wagner, A.D.: Cognitive control in media multitaskers. Proc. Nat. Acad. Sci. **106**(37), 15583–15587 (2009)
6. Ou, L.C., Luo, M.R., Woodcock, A., Wright, A.: A study of colour emotion and colour preference. Part I: colour emotions for single colours. Color Res. Appl. **29**(3), 232–240 (2004)
7. Ou, L.C., Luo, M.R., Woodcock, A., Wright, A.: A study of colour emotion and colour preference. Part II: colour emotions for two-colour combinations. Color Res. Appl. **29**(4), 292–298 (2004)
8. Ou, L.C., Luo, M.R., Woodcock, A., Wright, A.: A study of colour emotion and colour preference. Part III: colour preference modeling. Color Res. Appl. **29**(5), 381–389 (2004)
9. Palmer, S.E., Schloss, K.B.: An ecological valence theory of human color preference. Proc. Nat. Acad. Sci. **107**(19), 8877–8882 (2010)
10. Raudonis, V., Maskeliūnas, R., Stankevičius, K., Damaševičius, R.: Gender, age, colour, position and stress: How they influence attention at workplace? In: International Conference on Computational Science and Its Applications, pp. 248–264 (2017)
11. Schloss, K.B., Nelson, R., Parker, L., Heck, I.A., Palmer, S.E.: Seasonal variations in color preference. Cogn. Sci. **41**(6), 1589–1612 (2017). https://doi.org/10.1111/cogs.12429
12. Schloss, K.B., Palmer, S.E.: An ecological framework for temporal and individual differences in color preferences. Vis. Res. **141**, 95–108 (2017). https://doi.org/10.1016/j.visres.2017.01.010
13. Taylor, C., Schloss, K., Palmer, S.E., Franklin, A.: Color preferences in infants and adults are different. Psychon. Bull. Rev. **20**(5), 916–922 (2013)
14. Terwogt, M.M., Hoeksma, J.B.: Colors and emotions: preferences and combinations. J. Gen. Psychol. **122**(1), 5–17 (1995)
15. Zajonc, R.B.: Mere exposure: a gateway to the subliminal. Curr. Dir. Psychol. Sci. **10**(6), 224–228 (2001)
16. Zailskaitė-Jakštė, L., Ostreika, A., Jakštas, A., Stanevičienė, E., Damaševičius, R.: Brand communication in social media: the use of image colours in popular posts. In: Information and Communication Technology, Electronics and Microelectronics, pp. 1373–1378 (2017)
17. Yokosawa, K., Schloss, K.B., Asano, M., Palmer, S.E.: Ecological effects in cross-cultural differences between US and Japanese color preferences. Cogn. Sci. **40**(7), 1590–1616 (2016)

Asynchronous Client-Side Coordination of Cluster Service Sessions

Karolis Petrauskas(✉) and Romas Baronas

Vilnius University, Institute of Computer Science, Vilnius, Lithuania
{karolis.petrauskas,romas.baronas}@mif.vu.lt

Abstract. A system-to-system communication involving stateful sessions between a clustered service provider and a service consumer is investigated in this paper. An algorithm allowing to decrease a number of calls to failed provider nodes is proposed. It is designed for a clustered client and is based on an asynchronous communication. A formal specification of the algorithm is formulated in the TLA$^+$ language and was used to investigate the correctness of the algorithm.

Keywords: Session management · Cluster · Formal specification

1 Introduction

Nowadays business applications include a lot of interactions with service providers for handling various operations like order and payment processing, application monitoring and other specialized services [11]. The business applications themselves are often provided as services [12]. This kind of system architecture leads to many system-to-system integrations. Requirements for high availability and fault tolerance impose use of clustered topologies. In the case of system-to-system communications, clustered topologies are often used on the service consumer side as well as on the service provider side. The service providers are often deployed on a cloud or another virtualized infrastructure. Such infrastructure provides a lot of flexibility, but introduces a network instability, connection drops and other disruptions caused node migrations [1,14].

A lot of service providers implement the model of the eventual consistency in order to maintain high availability together with the service scalability [3]. That means the consistency is not guaranteed globally and special requirements are imposed on the service consumers in order to minimize the observed inconsistency. A common requirement for the clients of such services is to maintain session stickiness to particular nodes in the provider cluster [13]. That applies also to the stateless protocols, as requests for the particular end-user should be routed to the same back-end node in order to minimize the primary-node or the cache misses causing data inconsistency for a particular user.

A lot of mainstream protocols have no support for detecting lost connections or server failures immediately [2]. In such cases, the node availability should be

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 121–133, 2018.
https://doi.org/10.1007/978-3-319-97571-9_11

tracked by examining responses to the service requests. Only specific faults can be used as an indication of the failed provider node, excluding all the business faults as well as bad requests. If the service is accessed rarely, additional fake requests can be performed in order to keep the sessions alive or to detect node failures faster, before next user request will be received.

One of the ways to handle failing provider nodes is to consider another server from the remaining list and use it onwards for the session. This strategy can be inefficient if applied for each session separately, without sharing the knowledge on the failed nodes in the case of multiple sessions bound to a single provider node. After detecting the node failure, the error can be propagated to the caller or failover to another provider node can be performed silently, without interrupting the caller. Even if the error is handled by the consumer application, usually it has an impact on the behaviour of the system at least as increased execution time of some operations [2,8]. Because of that, the number of calls reaching the failed nodes should be minimized. The optimization usually includes sharing the node availability information between the sessions.

Applications consuming the provider services are often implemented as clusters themselves. The state sharing in the cluster is much more expensive than in a single node, especially if consistency should be preserved [12]. Inconsistency in tracking back-end availability has relatively low cost, as fixing it can only cause several unnecessary calls to the failed nodes. Keeping that in mind, it is reasonable to implement the sharing of the back-end node availability without consistency guarantees, employing the best-effort strategy. One of the ways for implementing it is to use asynchronous messages to share the known information on the provider availability.

Different applications require complex event processing relying on the detection of composite events often formed by logical and temporal combinations of events coming from many sources [9]. Various formal methods handling temporally composed events have been designed and implemented for complex event processing [4,7]. The Temporary Logic of Actions (TLA) is among such methods successfully used to describe behaviours of concurrent systems [5]. The corresponding specification language TLA$^+$ and the TLC model checker help to prevent serious bugs from reaching production as well as to optimize complex algorithms without sacrificing quality [10].

An algorithm for coordinating sessions using asynchronous messages in the consumer cluster is proposed in this paper. In order to avoid misbehaviours in various corner cases, the algorithm was formulated as a formal specification in the TLA$^+$ language [5,6]. The specification was verified by performing model checking [15], employing the TLC tool provided by the TLA$^+$ toolbox. We first provide a direct solution of the problem in Sect. 3 and show its misbehaviour by performing model checking. Then we propose two modifications of the algorithm in Sects. 4 and 5. In Sect. 6 we provide the details on the performed model checking and discuss the difference between the proposed correct solutions.

2 Principal Structure

We consider an interaction between two systems – a service provider and a service consumer. Both systems are assumed to be master-less clusters consisting of several nodes. The nodes of the consumer cluster maintain a set of sessions bound to some nodes in the service provider cluster. A structure of the elements participating in the session management is shown as a UML class diagram in Fig. 1.

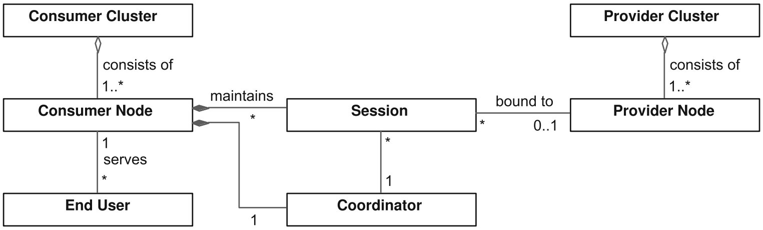


Fig. 1. Principal structure of the modelled subsystem

The main idea of the session management algorithm is that a client-side session process notifies its coordinator when a failure of the provider node is detected. The coordinator then notifies the other sessions bound to the same provider node and the coordinators on the other consumer nodes. The coordinators then notifies the corresponding sessions on their nodes. In that way, all the sessions in the cluster can handle the failure of the provider node gracefully.

We assume that a session can be bound to another node in the case of a provider failure, although re-binding of sessions should be avoided, as the cost of such operation is not negligible. The cost can be expressed in terms of performance drop or a possibility to provide the end-user with inconsistent data, etc. A session can be unbound, i.e. not bound to any of the provider nodes. This can be the case for the sessions that were dropped by the provider and were not reconnected yet.

3 Formal Specification

In this section we define a formal specification for the session management algorithm that relies on the asynchronous communication for sharing the knowledge about the provider node availability. We assume each session to be a separate process in a node. These processes communicate asynchronously with a coordinator process responsible for tracking a state of the provider cluster in the consumer node.

3.1 State of the Model

The specification of the session management algorithm is formulated in the TLA$^+$ language and has several parameters (constants). A constant in the specification does not change during a single simulation (model checking), but can have different values in separate simulations. The following excerpt defines constants and a state structure of the proposed specification:

$$
\begin{array}{l}
\textsc{constants}\ \mathit{PNodes},\ \mathit{CNodes},\ \mathit{SNames}\\
\textsc{variables}\ \mathit{prov},\ \mathit{cons}\\
\mathit{NA}\ \triangleq\ \textsc{choose}\ n : n \notin \mathit{PNodes}\\
\mathit{Msg}\ \triangleq\ [\mathit{pn} : \mathit{PNodes}]\\
\mathit{TypeOK}\ \triangleq\ \mathit{prov} \in [\mathit{PNodes} \rightarrow \textsc{boolean}\ ] \land \mathit{cons} \in [\mathit{CNodes} \rightarrow [\\
\qquad\qquad\qquad\qquad c : [\mathit{PNodes}\ \rightarrow [\mathit{st}\ \ : \textsc{boolean}\ ]],\\
\qquad\qquad\qquad\qquad s : [\mathit{SNames} \rightarrow [\mathit{pn} : \mathit{PNodes} \cup \{\mathit{NA}\},\ m : \textsc{subset}\ \mathit{Msg}]],\\
\qquad\qquad\qquad\qquad \mathit{sm} : \textsc{subset}\ \mathit{Msg},\ \mathit{cm} : \textsc{subset}\ \mathit{PNodes}]]
\end{array}
$$

In order to keep the specification simple and the state space finite, we consider a number of consumer and provider nodes as well as a number of sessions in each node to be constant. The constant *PNodes* stands for a set of provider nodes. Each node in this set is defined by assigning a unique identifier, e.g. $\mathit{PNodes} = \{\mathrm{p}_1, \mathrm{p}_2\}$. Similarly the constant *CNodes* stands for a set of consumer nodes. The constant *SNames* stands for a session pool in the consumer node and should be assigned with a set of session identifiers.

Systems are modelled as state machines in TLA$^+$. Variables define a state structure of the machine. In this specification the variable *prov* represents the actual state of the provider nodes. This variable is a function with the domain *PNodes* and the range BOOLEAN, where TRUE means the corresponding node is operational, and FALSE – the node is down.

The variable *cons* represents the state of the consumer cluster including its view of the provider nodes. It is a function with a domain *CNames* and therefore describes state for each node in the consumer cluster separately.

A state of the coordinator process is represented by the field c, that holds known states for all the provider nodes in each consumer node. The state of a particular provider node $\mathit{cons}[\mathit{cn}].c[\mathit{pn}].\mathit{st}$ (where $\mathit{cn} \in \mathit{CNodes}$ and $\mathit{pn} \in \mathit{PNodes}$) can differ from $\mathit{prov}[\mathit{pn}]$, because changes of the node availability are not detected immediately by the consumer nodes.

The field $\mathit{cons}[\mathit{cn}].s$ stands for a session pool in a consumer node. Each session $\mathit{cons}[\mathit{cn}].s[\mathit{sn}]$ (where $\mathit{sn} \in \mathit{SNames}$) is bound to a node $\mathit{pn} \in \mathit{PNodes}$ or is unbound, if $\mathit{cons}[\mathit{cn}].s[\mathit{sn}].\mathit{pn} = \mathit{NA}$. The session has also a set of asynchronous messages $\mathit{cons}[\mathit{cn}].s[\mathit{sn}].m$ received from the coordinator in the current node. Synchronous calls are modelled as direct changes of the corresponding variables. In this algorithm we consider messages sent to the sessions by the coordinator to be asynchronous. The set of possible messages is defined as *Msgs*.

The fields *sm* and *cm* in $\mathit{cons}[\mathit{cn}]$ stand for sets of asynchronous messages received by the coordinator process correspondingly from the sessions in the current node and the coordinators in other consumer nodes.

A set of valid states in the specification is defined by the predicate *TypeOK*. This predicate is used to check the type correctness of the specification.

3.2 Behaviour of the Provider Nodes

Transitions of the state machine are defined by the actions – formulas involving primed variables (they stand for the variable values in the next step). Actions describing behaviour of the provider nodes are the following:

$$\begin{aligned}
&\mathit{ProvNodeUp}(pn) \triangleq \neg \mathit{prov}[pn]\\
&\quad \land \mathit{prov}' = [\mathit{prov}\ \textsc{except}\ ![pn] = \textsc{true}]\\
&\quad \land \textsc{unchanged}\ \langle \mathit{cons} \rangle\\
&\mathit{ProvNodeDown}(pn) \triangleq \mathit{prov}[pn]\\
&\quad \land \mathit{prov}' = [\mathit{prov}\ \textsc{except}\ ![pn] = \textsc{false}]\\
&\quad \land \textsc{unchanged}\ \langle \mathit{cons} \rangle
\end{aligned}$$

The action $\mathit{ProvNodeUp}(pn)$ states that the provider node $pn \in \mathit{PNodes}$ can become operational at any time if it is currently down. The expression $[\mathit{prov}\ \textsc{except}\ ![pn] = \textsc{true}]$ stands for a function that is equal to *prov* except that the value of $\mathit{prov}[pn]$ equals to TRUE. The state of the consumer nodes is not affected by this transition ($\textsc{unchanged}\ \langle \mathit{cons} \rangle$) as the availability of the provider node is only detected by the consumer later by performing some operations. The action $\mathit{ProvNodeDown}(pn)$ correspondingly turns operational node down.

3.3 Behaviour of a Consumer Session

This section describes operation of the session processes. A session can either handle requests, update its state based on messages from the coordinator or connect if it was not bound to any provider node. The latter is modelled by the action $\mathit{SessionConnect}(cn, sn)$, where $cn \in \mathit{CNodes}$ stands for a consumer node and $sn \in \mathit{SNames}$ stands for a session identifier. This action is enabled, if the session is not bound to a provider node ($\mathit{cons}[cn].s[sn] = \mathit{NA}$) and there is a node $pn \in \mathit{PNodes}$ that is operational ($\mathit{prov}[pn] = \textsc{true}$) and the consumer node knows it is operational ($\mathit{cons}[cn].c[pn].st = \textsc{true}$),

$$\begin{aligned}
&\mathit{SessionConnect}(cn,\ sn) \triangleq \mathit{cons}[cn].s[s].pn = \mathit{NA} \land \mathit{cons}[cn].c[pn].st\\
&\quad \land \exists\, pn \in \mathit{PNames} :\ \land \mathit{prov}[pn]\\
&\qquad\qquad\qquad\qquad\quad \land \mathit{cons}' = [\mathit{cons}\ \textsc{except}\ ![cn].s[sn].pn = pn]\\
&\qquad\qquad\qquad\qquad\quad \land \textsc{unchanged}\ \langle \mathit{prov} \rangle
\end{aligned}$$

When connected ($\mathit{cons}[cn].s[sn].pn \in \mathit{PNodes}$), a session can be used by the consumer node to issue requests to the service provider. Only failing requests are modelled in this specification, because the successful requests do not affect the state of the modelled subsystem. We consider all the requests ended up with business faults as completed successfully. A request is considered failed only if the corresponding provider node is down ($\mathit{prov}[pn] = \textsc{false}$) at the moment, when the request is performed. In that case the session marks itself as unbound and sends an asynchronous message indicating the failure of the provider node to the coordinator process. The state of the other sessions as well as the state of the coordinator is not affected in this transition directly,

$$\begin{aligned}
&\mathit{SessionReqFail}(cn,\, sn) \triangleq\ \mathit{cons}[cn].s[sn].pn \in \mathit{PNodes} \land \neg \mathit{prov}[\mathit{cons}[cn].s[sn].pn] \\
&\quad \land \mathit{cons}' = [\mathit{cons}\ \text{EXCEPT} \\
&\qquad\qquad\qquad !\,[cn].s[sn].pn = \mathit{NA}, \\
&\qquad\qquad\qquad !\,[cn].sm = @ \cup \{[pn \mapsto \mathit{cons}[cn].s[sn].pn]\}] \\
&\quad \land \text{UNCHANGED}\ \mathit{prov}
\end{aligned}$$

The symbol @ in this and other formulas stands for the current value of the function.

Sending an asynchronous message is modelled by adding it to the set of messages $\mathit{cons}[cn].sm$ sent by the sessions to the coordinator. The ordering of messages is not modelled in this specification in order to decrease the space of possible states. This also allows to avoid complicated requirements for an implementation. Duplicated messages are modelled by not removing a message from the set $\mathit{cons}[cn].sm$ after processing it.

A session can receive notifications from the coordinator indicating provider nodes that became down. Upon receiving such a message the session unbinds itself, if the provider node specified in the message matches with the node the session is bound to. This behaviour is modelled by the following action:

$$\begin{aligned}
&\mathit{SessionUpdate}(cn,\, sn) \triangleq \\
&\quad \exists\, \mathit{msg} \in \mathit{cons}[cn].s[sn].m : \\
&\qquad \exists\, \mathit{msgsDeq} \in \{\mathit{cons}[cn].s[sn].m,\ \mathit{cons}[cn].s[sn].m \setminus \{\mathit{msg}\}\} : \\
&\qquad\quad \land \mathit{cons}' = \text{LET}\ \mathit{consDeq} \triangleq [\mathit{cons}\ \text{EXCEPT}\ !\,[cn].s[sn].m = \mathit{msgsDeq}] \\
&\qquad\qquad\qquad\quad \text{IN}\ \ \text{IF}\ \mathit{msg}.pn = \mathit{cons}[cn].s[sn].pn \\
&\qquad\qquad\qquad\qquad\quad \text{THEN}\ [\mathit{consDeq}\ \text{EXCEPT}\ !\,[cn].s[sn].pn = \mathit{NA}] \\
&\qquad\qquad\qquad\qquad\quad \text{ELSE}\ \ \mathit{consDeq} \\
&\qquad\quad \land \text{UNCHANGED}\ \mathit{prov}
\end{aligned}$$

Receiving a message (dequeuing) is modelled by taking any message from the set of sent messages $\mathit{cons}[cn].s[sn].m$ ignoring their order. The set of sent messages is either left unchanged or the selected message is removed from that set. The former case models the situation when there were several identical messages in the queue. This case also models duplication of messages, that can occur because of various retries or message re-deliveries in the software implementing this algorithm. The latter case models dequeuing of the last message of that kind. It also covers the situation, when part of messages can be lost.

3.4 Behaviour of the Consumer Node Coordinator

The coordinator is responsible for maintaining the state of the provider nodes in a single consumer node. It is responsible also for sharing this knowledge between the consumer nodes. The coordinator receives messages indicating failures of the provider nodes from the sessions. Then it updates its internal state ($\mathit{cons}[cn].c[pn].st$) and notifies all the sessions and other consumer nodes about the state changes, if some node becomes unavailable. This is modelled by the following action:

$$
\begin{array}{l}
\mathit{CoordSessionMsg}(cn) \triangleq \\
\quad \exists\, msg \in cons[cn].sm : \exists\, sm \in \{cons[cn].sm,\ cons[cn].sm \setminus \{msg\}\} : \\
\quad\quad \text{LET}\ \ consDeq \triangleq [cons\ \text{EXCEPT}\ ![cn].sm = sm] \\
\quad\quad\quad\quad\quad consEnq \triangleq [c \in \text{DOMAIN}\ consDeq \mapsto [consDeq[c]\ \text{EXCEPT} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad !.cm = \text{IF}\ c = cn\ \text{THEN}\ @\ \text{ELSE}\ @ \cup \{msg.pn\}]] \\
\quad\quad\quad\quad\quad consUpd \triangleq [consEnq\ \text{EXCEPT} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad ![cn].c[msg.pn].st = \text{FALSE}, \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad ![cn].s = [s \in \text{DOMAIN}\ @ \mapsto [@[s]\ \text{EXCEPT}\ !.m = @ \cup \{msg\}]]] \\
\quad\quad \text{IN}\quad \wedge\ cons' = \text{IF}\ cons[cn].c[msg.pn].st\ \text{THEN}\ consUpd\ \text{ELSE}\ consDeq \\
\quad\quad\quad\quad\quad \wedge\ \text{UNCHANGED}\ prov
\end{array}
$$

The coordinator sends notifications to other consumer nodes when some provider node becomes offline. These notifications are handled by coordinators on the corresponding consumer nodes. Handling these notifications is modelled by the action *CoordClusterMsg*(cn). Main distinction from *CoordSessionMsg*(cn) in this action is that the corresponding provider node is checked explicitly (the conjunct $\neg prov[pn]$) before marking it as offline. This check is kept explicit in order to avoid accidental marking of operational node as unavailable. We assume the network can make a particular provider node visible from one consumer node and not visible from another. Another distinction is that the notification is not propagated to other nodes here,

$$
\begin{array}{l}
\mathit{CoordClusterMsg}(cn) \triangleq \\
\quad \exists\, pn \in cons[cn].cm : \exists\, cm \in \{cons[cn].cm,\ cons[cn].cm \setminus \{pn\}\} : \\
\quad\quad \text{LET}\ \ consDeq \triangleq [cons\ \text{EXCEPT}\ ![cn].cm = cm] \\
\quad\quad\quad\quad\quad\ consEnq \triangleq [consDeq\ \text{EXCEPT}\ ![cn].c[pn].st = \text{FALSE}, \\
\quad\quad\quad\quad\quad\quad ![cn].s = [s \in \text{DOMAIN}\ @ \mapsto [@[s]\ \text{EXCEPT}\ !.m = @ \cup \{[pn \mapsto pn]\}]]] \\
\quad\quad \text{IN}\quad\ \wedge\ cons' = \text{IF}\ cons[cn].c[pn].st \wedge \neg prov[pn]\ \text{THEN}\ consEnq\ \text{ELSE}\ consDeq \\
\quad\quad\quad\quad\quad\ \wedge\ \text{UNCHANGED}\ prov
\end{array}
$$

As shown above, the coordinator marks the provider nodes as being down in the consumer state based on messages from the sessions and the other coordinators. The coordinator is also responsible for marking the nodes as being available, when they become operational. This is performed periodically by checking the nodes that are currently marked as down ($cons[cn].c[pn].st = \text{FALSE}$) and marking them available if the checks succeed. This is modelled by the action *CoordProviderCheck*(cn, pn). The check of the provider node is performed synchronously and is modelled here by the conjunct $prov[pn]$,

$$
\begin{array}{l}
\mathit{CoordProviderCheck}(cn,\ pn) \triangleq \neg cons[cn].c[pn].st \wedge prov[pn] \\
\quad \wedge\ cons' = [cons\ \text{EXCEPT}\ ![cn].c[pn].st = \text{TRUE}] \\
\quad \wedge\ \text{UNCHANGED}\ prov
\end{array}
$$

3.5 Temporal Properties

The complete specification in TLA$^+$ is represented as a temporal formula

$$\mathit{Spec} \triangleq \mathit{Init} \wedge \Box[\mathit{Next}]_{\langle \mathit{prov}, \mathit{cons} \rangle} \wedge \mathit{Liveness}$$

where *Init* describes the initial state, *Next* defines all the possible transitions at any step and *Liveness* defines requirements for actions to actually occur. Here $\Box$ is a temporal operator "always". The expression $[\mathit{Next}]_{\langle \mathit{prov}, \mathit{cons} \rangle}$ states that either a step *Next* or a step not changing the variables *prov* and *cons* can occur.

The formula *Init* stands for the initial state. It is similar to the *TypeOK* predicate, except that message sets are initialized with empty sets $\{\}$ and all the provider nodes are assumed to be operational initially. The formula *Next* is a disjunction of all the actions and describes all the possible transitions at any step. This formula straightforward and therefore is omitted in this paper.

Liveness is a temporal formula describing what actions should actually occur in the system if they are enabled (contrary to "can occur"). We assume weak fairness conditions (an action will be performed if it is enabled forever) for all the actions describing behaviour of the consumer nodes (the sessions and the coordinators).

The specification *Spec* can be used to check if it satisfies required properties. A typical property usually checked for any specification is a type correctness invariant

$$\mathit{TypeInvariant} \triangleq \mathit{Spec} \Rightarrow \Box\, \mathit{TypeOK}$$

Apart from simple invariants, TLA$^+$ allows to define temporal properties. These properties imply requirements for the entire behaviour (a sequence of transitions). The following temporal properties are expected to be held in the system:

$$\begin{array}{l}
\mathit{NodeDownDetected} \triangleq \\
\quad \forall\, \mathit{pn} \in \mathit{PNodes},\ \mathit{cn} \in \mathit{CNodes},\ \mathit{sn} \in \mathit{SNames} : \\
\qquad (\mathit{cons}[\mathit{cn}].s[\mathit{sn}].\mathit{pn} = \mathit{pn} \wedge \neg \mathit{prov}[\mathit{pn}]) \leadsto (\mathit{cons}[\mathit{cn}].s[\mathit{sn}].\mathit{pn} = \mathit{NA} \vee \mathit{prov}[\mathit{pn}]) \\
\mathit{SessionsWillReconnect} \triangleq \\
\quad \forall\, \mathit{pn} \in \mathit{PNodes},\ \mathit{cn} \in \mathit{CNodes},\ \mathit{sn} \in \mathit{SNames} : \\
\qquad (\mathit{cons}[\mathit{cn}].s[\mathit{sn}].\mathit{pn} = \mathit{NA} \wedge \mathit{prov}[\mathit{pn}]) \leadsto (\mathit{cons}[\mathit{cn}].s[\mathit{sn}].\mathit{pn} \neq \mathit{NA} \vee \neg \mathit{prov}[\mathit{pn}])
\end{array}$$

The temporal property *NodeDownDetected* asserts that if a provider node becomes unavailable, then sessions bound to it will be eventually disconnected, unless the node will become operational again ($\leadsto$ is the temporal operator "leads to"). It was checked that this property holds for the specification by employing the TLC model checker.

The property *SessionsWillReconnect* asserts, that if a session is unbound and there is an operational node, the session will reconnect and will continue to serve requests. These properties should be implied by the specification,

$$\mathit{TemporalProperties} \triangleq \mathit{Spec} \Rightarrow \mathit{NodeDownDetected} \wedge \mathit{SessionsWillReconnect}$$

The TLC model checker was used to check the type correctness invariant as well as the temporal properties defined above. The model checking showed that property *SessionsWillReconnect* is not satisfied by the specification. The misbehaviour is caused by the asynchronous communication between the sessions and the coordinator. One of the counter-examples: a provider node was down, then it becomes available, coordinator process marks it as available and then receives a delayed message from a session indicating node failure. As a consequence, the node is marked as unavailable again till the next $CoordProviderCheck(pn)$. This behaviour can repeat infinitely, making the consumer to consider running provider node as failed thus decreasing availability of the system.

4 Explicit Provider Checks

Another possible solution allowing to avoid the impact of the delayed messages is to check node availability before marking it as offline in the coordinator process. In that case, the $CoordSessionMsg(cn)$ action should be changed by adding expression $cons[cn].c[msg.pn].st \land \neg prov[msg.pn]$ instead of $cons[cn].c[msg.pn].st$ in the IF condition. The changed parts of the action are as follows:

$$
\begin{aligned}
&CoordSessionMsg(cn) \triangleq \\
&\quad \ldots \\
&\quad \land\ cons' = \text{IF } cons[cn].c[msg.pn].st \land \neg prov[msg.pn] \text{ THEN } consUpd \text{ ELSE } consDeq \\
&\quad \land\ \text{UNCHANGED } prov
\end{aligned}
$$

With this change the temporal property *SessionsWillReconnect* is fulfilled. The drawback of this approach is that additional calls to the service provider must be performed.

5 Detecting Delayed Messages

In order to avoid the impact of the delayed messages, generations of the provider nodes can be introduced. Each time when a provider node is detected to become online by the coordinator, its generation number is increased. Messages referring to generations older than one known by the coordinator are then ignored. The generations should be tracked in the coordinator process as well as in the sessions and should be included in the messages exchanged between them. The updated structure of the messages and the state of the model are as follows:

$$
\begin{aligned}
&Msg \triangleq [pn : PNodes,\ gen : Nat] \\
&TypeOK \triangleq \\
&\quad \land\ prov \in [PNodes \rightarrow \text{BOOLEAN}\ ] \\
&\quad \land\ cons \in [CNodes \rightarrow [\\
&\qquad\quad c : [PNodes\ \rightarrow [st\ \ : \text{BOOLEAN}\ ,\ gen : Nat]], \\
&\qquad\quad s : [SNames \rightarrow [pn : PNodes \cup \{NA\},\ gen : Nat,\ m : \text{SUBSET } Msg]], \\
&\qquad\quad sm : \text{SUBSET } Msg,\ cm : \text{SUBSET } PNodes]]
\end{aligned}
$$

The observed generations of the provider nodes are tracked inside of the consumer nodes and are not shared between them. Each node can observe different provider node interruptions. Moreover, depending on a network topology, a particular provider node can be accessible from one consumer node and not accessible from other. The message delays between the consumer nodes are handled by the explicit node checks (conjunct $\neg prov[pn]$) in the action $\mathit{CoordClusterMsg}(cn)$.

Some parts of the model should be updated to maintain the observed provider node generations. The initial state can start from any generation. We consider to have $gen \mapsto 0$ in all the sessions and the coordinators.

For the coordinator behaviour, the $\mathit{CoordProviderCheck}(cn, pn)$ action is changed to increment the node generation each time the coordinator detects it became available,

$$\begin{array}{l} \mathit{CoordProviderCheck}(cn,\ pn) \triangleq \neg \mathit{cons}[cn].c[pn].st \land \mathit{prov}[pn] \\ \quad \land \mathit{cons}' = [\mathit{cons}\ \textsc{except}\ ![cn].c[pn].st = \textsc{true},\ \ ![cn].c[pn].gen = @ + 1] \\ \quad \land \textsc{unchanged}\ \mathit{prov} \end{array}$$

The coordinator then ignores all the messages received with old generations ($msg.gen < \mathit{cons}[cn].c[msg.pn].gen$) in the $\mathit{CoordSessionMsg}(cn)$ action. It also includes the generation into the messages sent to the sessions when node change is detected on a notification from other consumer nodes in $\mathit{CoordClusterMsg}(cn)$. The generation is included in the messages triggered by the session notifications in the $\mathit{CoordSessionMsg}(cn)$ action without changes in the specification as it only forwards received messages (and they include the gen field).

When connecting, a session takes the current provider node generation from the coordinator in the consumer node ($\mathit{cons}[cn].c[pn].gen$), therefore the action $\mathit{SessionConnect}(cn, sn)$ is updated to assign the generation known by the session as follows:

$$\begin{array}{lll} \mathit{SessionConnect}(cn,\ sn) & \triangleq & \mathit{cons}[cn].s[s].pn = \mathit{NA} \land \mathit{cons}[cn].c[pn].st \\ \quad \land \exists\, pn \in \mathit{PNames}: & & \land\ \mathit{prov}[pn] \\ & & \land\ \mathit{cons}' = [\mathit{cons}\ \textsc{except}\ \ ![cn].s[sn].pn = pn, \\ & & \qquad\qquad\qquad\qquad\qquad ![cn].s[sn].gen = \mathit{cons}[cn].c[pn].gen] \\ & & \land\ \textsc{unchanged}\ \langle \mathit{prov} \rangle \end{array}$$

The sessions should only consider messages received from the coordinator in the $\mathit{SessionUpdate}(cn, sn)$ action with a generation not less than the current generation known by the session ($\mathit{cons}[cn].s[s].gen \leq msg.gen$) and then remember it as the last known generation ($![cn].s[s].gen = msg.gen$). All the other messages are just dequeued and ignored. The sessions include the generation to the messages sent to the coordinator in the $\mathit{SessionReqFail}(cn, sn)$ action.

6 Model Checking

Three variants of the specification were defined in Sects. 3, 4 and 5. All of them were model-checked for the type correctness as well as for the temporal properties. The model checking was performed on a laptop with 8 CPUs, 16 GB RAM and an SSD disk, using TLA$^+$ Toolbox version 1.5.6 with OpenJDK Java version 1.8.0 running on a Linux OS.

A model checking in TLA$^+$ is performed by defining a model and exploring its possible states. The model instantiates the specification with particular values for the constants. In all the cases the following values were used for the specification constants:

$$\mathit{CNodes} = \{\mathrm{c}_1, \mathrm{c}_2\}, \mathit{PNodes} = \{\mathrm{p}_1, \mathrm{p}_2\}, \mathit{SNames} = \{\mathrm{s}_1, \mathrm{s}_2\}.$$

Small sets were selected in order to decrease state space while keeping the model meaningful.

The specification maintaining observed generations of the provider nodes (Sect. 5) has fields $gen \in \mathit{Nat}$, whose range is infinite. In order to keep the state space finite, additional constraint was used when checking the model,

$$\forall cn \in \mathit{CNodes}, pn \in \mathit{PNodes} : \mathit{cons}[cn].c[pn].gen < 3.$$

For all the variants of the specification the type correctness invariant ($\mathit{Spec} \Rightarrow \Box\, \mathit{TypeOK}$) was checked for the entire space of the state values (with the constraints shown above). The temporal properties were checked only for a fraction of the state space. If a model checking was running for 10 h with no property violations reported, the checking was stopped and considered successful. The violation of the *SessionsWillReconnect* property for the initial specification (Sect. 3) was found in 3 min.

6.1 Modelling Message Queues

Message queues were modelled as sets of sent messages [6]. This approach ignores the order of messages and also cannot represent several identical messages in the queue precisely.

Another way to model a message queue is to use a sequence instead of a set. In that case the order of messages is maintained and multiple messages with the same representation are supported. While this approach is more precise, it increases the state space a lot. The corresponding specifications were designed during this research with message queues represented by sequences. The model checker was unable to explore all the state space (the checking was stopped) even for the type correctness invariant with a constraint for the queues to have length less than 3.

Note also, that the violation of the property *SessionsWillReconnect* was not reproduced with these models in a reasonable time. Possible reasons for not reproducing the violation are: the model was to small (maybe longer message queues should be considered), the model checking was cancelled to early, respect of the message order solves the race condition leading to violation of this property. Particular reason for this was not found in this research.

6.2 Number of Synchronous Provider Checks

The specification variants in Sects. 4 and 5 both solve the race condition found when model checking the initial specification (Sect. 3).

The specification with the explicit checks (Sect. 4) is simpler to implement, though is not efficient. A session needs to check the provider node availability by performing a synchronous call each time it receives notification from the coordinator. The number of calls will be equal to the number of sessions bound to that particular provider node. In the case of uniform distribution of sessions over the provider nodes there will be $|SNames| \times |CNodes|/|PNodes|$ checks performed (usually $|SNames| \gg |PNodes|$). In the worst case (all the sessions are be bound to a single node) the number of checks will be $|SNames| \times |CNodes|$. The number of synchronous checks can cause two kinds of problems:

1. If the provider node is actually online, unnecessary calls will be issued to the provider. All the checks will be performed approximately in the same time and therefore can cause a notable increase in a load on the provider. The sessions cannot serve user requests while performing the check.
2. If the provider node is not reachable, the checks can take long time before failing, causing delays before switching to another node. This can be the case, if the provider becomes unavailable because of network interruptions or misconfiguration, e.g. when packets are dropped instead of rejecting them.

The specification maintaining the observed node generations (Sect. 5) allows to decrease the number of calls to the failing nodes. The number of synchronous checks will be equal to the number of nodes in the consumer cluster, as each node will perform single synchronous check.

7 Conclusions

The proposed algorithm for tracking provider node availability allows to avoid synchronous communication in the consumer cluster as well as inside of the consumer node. That allows to avoid process blocking thus decreasing impact on the performance. The algorithm was formulated by employing formal specification language and was model-checked for its correctness in a subset of its possible states.

The model checking showed that straight-forward solution of the problem works incorrectly at some race-conditions. Explicit node checks can be used to solve the inconsistencies though they introduce a lot of overhead and can cause bottlenecks in the system. The overhead can be decreased by tracking observed generations of the provider nodes. It is meaningful to track the generations in a single consumer node, although its usefulness cluster-wide depend on the network topology.

References

1. Armbrust, M., et al.: A view of cloud computing. Commun. ACM **53**(4), 50–58 (2010). https://doi.org/10.1145/1721654.1721672
2. Ayari, N., Barbaron, D., Lefevre, L., Primet, P.: Fault tolerance for highly available internet services: concepts, approaches, and issues. IEEE Commun. Surv. Tutor. **10**(2), 34–46 (2008). https://doi.org/10.1109/COMST.2008.4564478

3. Bailis, P., Ghodsi, A.: Eventual consistency today: limitations, extensions, and beyond. Queue **11**(3), 20:20–20:32 (2013). https://doi.org/10.1145/2460276.2462076
4. Hinze, A., Voisard, A.: EVA: an event algebra supporting complex event specification. Inf. Syst. **48**, 1–25 (2015). https://doi.org/10.1016/j.is.2014.07.003
5. Lamport, L.: The temporal logic of actions. ACM Trans. Program. Lang. Syst. **16**(3), 872–923 (1994). https://doi.org/10.1145/177492.177726
6. Lamport, L.: Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers. Addison-Wesley Longman Publishing Co., Inc., Boston (2002)
7. Li, D., Zhang, Q., Zio, E., Havlin, S., Kang, R.: Network reliability analysis based on percolation theory. Reliab. Eng. Syst. Saf. **142**, 556–562 (2015). https://doi.org/10.1016/j.ress.2015.05.021
8. Lowell, D.E., Chandra, S., Chen, P.M.: Exploring failure transparency and the limits of generic recovery. In: Proceedings of the 4th Conference on Symposium on Operating System Design & Implementation, vol. 4, p. 15, OSDI 2000, USENIX Association, Berkeley (2000). Article No. 20
9. Luckham, D.C.: Event Processing for Business: Organizing the Real-Time Enterprise. Wiley, Hoboken (2015). https://doi.org/10.1002/9781119198697
10. Newcombe, C., Rath, T., Zhang, F., Munteanu, B., Brooker, M., Deardeuff, M.: How amazon web services uses formal methods. Commun. ACM **58**(4), 66–73 (2015). https://doi.org/10.1145/2699417
11. Petcu, D.: Consuming resources and services from multiple clouds. J. Grid Comput. **12**(2), 321–345 (2014). https://doi.org/10.1007/s10723-013-9290-3
12. Tsai, W., Bai, X., Huang, Y.: Software-as-a-service (saas): perspectives and challenges. Sci. Chin. Inf. Sci. **57**(5), 1–15 (2014). https://doi.org/10.1007/s11432-013-5050-z
13. Vogels, W.: Eventually consistent. Commun. ACM **52**(1), 40–44 (2009). https://doi.org/10.1145/1435417.1435432
14. Xie, R., Wen, Y., Jia, X., Xie, H.: Supporting seamless virtual machine migration via named data networking in cloud data center. IEEE Trans. Parallel Distrib. Syst. **26**(12), 3485–3497 (2015). https://doi.org/10.1109/TPDS.2014.2377119
15. Yu, Y., Manolios, P., Lamport, L.: Model checking TLA$^+$ specifications. In: Pierre, L., Kropf, T. (eds.) CHARME 1999. LNCS, vol. 1703, pp. 54–66. Springer, Heidelberg (1999). https://doi.org/10.1007/3-540-48153-2_6

Ping-Pong Tests on Distributed Processes Using Java Bindings of Open-MPI and Java Sockets with Applications to Distributed Database Performance

Mehmet Can Boysan(✉)

Institute of Computer Science, J. Liivi 2, 50409 Tartu, Estonia
mehmet.can.boysan@ut.ee

Abstract. The use of distributed database solutions is becoming more widespread due to their higher performance and storage capabilities compared to relational databases. Since these systems rely heavily on interprocess communications, an investigation on the effect of network latency is needed. In this paper, we examine the Java bindings of Open-MPI library running on InfiniBand and TCP/IP stack and the Java Socket API for TCP/IP communications with a simple ping-pong test with analysis of latency on performance of distributed in-memory key-value stores that operate in single data centers.

Keywords: Distributed databases · Network latency · Ethernet
InfiniBand · Java sockets · TCP · MPI · Java

1 Introduction

Distributed in-memory key-value stores are becoming more widespread to be used as the preferred caching solutions because of their superior performance and higher scalability compared to relational databases. However, the distributed nature of such systems require intensive inter-process communication to be able to provide acceptable levels of consistency, availability and fault-tolerance. One way to achieve these properties is by using consensus protocols to decide on the valid state of the system. This requires a lot of message passing between the processes. Therefore, measuring the communication latency is important.

In this paper, the Java bindings of the Open-MPI library and Java Sockets have been used to develop a program that can send Ping-Pong messages between the processes to compare communication latencies on InfiniBand and 10 Gb Ethernet interconnects. The motivation behind this study is to provide some results for communities that seek high performance and specifically want to use Java as implementation language.

The paper is structured as follows. First, a background and an analysis of the existing literature is given in Sect. 2. In the next Section, the testing environment and the methodology used for latency evaluation is described. The collected results are analyzed in Sect. 4 and the report is finalized with a conclusion.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 134–141, 2018.
https://doi.org/10.1007/978-3-319-97571-9_12

2 Background and Literature Review

NoSQL databases is being preferred instead of the relational databases since they can be deployed in a distributed fashion which allows them to scale horizontally when more power and performance is needed. One possibility is to use such systems as distributed in-memory key-value stores that are able to serve as high performing caching solutions. Some popular examples for such systems are Apache Ignite [5] and Hazelcast [1].

When such systems are deployed in a distributed manner however, they need ways to provide high availability, consistency and fault tolerance. Brewer's theorem on the other hand suggests that achieving them in case of a system failure is impossible [10]. Therefore, most of these systems apply some known consensus protocols like two/three phase commit (2/3PC) [12,18], Paxos [15] and Raft [17]. The problem with this approach is that many messages of relatively small sizes need to be passed between the connected processes in order to reach an agreement on the valid state of the overall system. On slow networks, as the number of processes increase, the overall performance would be negatively affected because of the added overhead of these message transmissions.

In order to mitigate such a performance degradation, different interconnects such as InfiniBand can be chosen instead of 10 Gigabit+ Ethernet. There have been some studies conducted such as [11,13] that compare these technologies by doing point-to-point communication benchmarks which show that InfiniBand outperforms 10 Gigabit Ethernet in terms of communication latency in the tested High Performance Computing (HPC) environments. As for the interprocess communications, any parallel communication library or language such as PCJ [16] or Titanium [20] can be chosen, but we prefer to stick with the well-established Message Passing Interface (MPI) [2] as the message passing solution.

Today, large vendors like Amazon provide highly scalable cloud infrastructures that use 10 Gb+ Ethernet instead of InfiniBand interconnect which do not provide better performance than a typical mid-range Linux cluster [14]. In some cases, setting up an in-house cluster might not be feasible, hence, sticking to a plan offered by such vendors might still be the preferred way for the businesses to fulfill their requirements. This means that such systems need to rely on the 10 Gb+ Ethernet interconnect which might prevent the system from reaching its full potential.

Another point in this discussion is that the distributed database solutions that use Java as their implementation language might not take full advantages of the native C implementations of the MPI communication standard. We have identified a number of Java implementations of this interface namely, mpiJava [3], MPJ Express [4] and also found that the Open-MPI distributions started including the Java bindings of their C implementations [19]. Although, some latency analysis comparing Java Open-MPI bindings with the original implementation is done, there seems to exist no literature that compares any MPI Java implementation with Java's TCP/IP socket layer in terms of communication latency. Hence, this paper intends to fill this gap by providing some experiments in this regard.

3 Test Environment

3.1 Test Hardware and Software

The tests were done on the *Rocket cluster* located in the High Performance Computer Center of University of Tartu [7]. Table 1 illustrates the hardware properties of a single compute node in the cluster. At most 10 compute nodes (out of 135) were used in the tests without any modifications on their existing hardware or software. Each compute node uses CentOS 7.4 as its operating system. Open-MPI version 1.8.4 and Oracle Java 8 with development kit (JDK) and runtime environment (JRE) version 1.8.0_25 are used.

Table 1. Hardware specifications of a single compute node in the Rocket cluster

CPU	2xIntel(R) Xeon(R) CPU E5-2660 v2 @ 2.20GHz (20 cores total) (4 CPU cores used in tests)
RAM	64 GB RAM
Storage	1TB HDD (860 GB usable)
Network	4x QDR Infiniband, 8 Mellanox switches
	10Gbit/s Ethernet, ConnectX-3 MT27500 Mellanox switch

3.2 Methodology

The software created for this investigation can be found in [9].

Ping-Pong Test Setup: In the code, a distributed process object, referred to as a `Role` was created. A `Role` can be thought of as a single processing unit and is supposed to be deployed to different cluster nodes. In this case, it is created to serve ping-pong messages. In a cluster of N `Roles`, a single leader is selected which sends a ping message to all the `Roles` in the cluster including itself. The time starting from the *first ping* and ending with the *last pong* gives the round trip latency of a single leader trying to send a consensus message. Individual ping-pong message round trip latencies between the `Roles` were also recorded.

Each `Role` uses Java's Socket API for TCP/IP communication, and `MPI.iRecv` routine for MPI's InfiniBand and TCP/IP communications. A single message listener accepts each connection in a loop and once a message arrives, a separate thread processes it. Messages are sent in a non-blocking manner, meaning that each message is sent in a dedicated thread without waiting for its completion. Messages are sent with Java sockets for TCP/IP and for MPI, with `MPI.iSend` routine.

The Ping and Pong messages are initially constructed as Java objects which are first marshalled into JSON strings and then converted into byte arrays prior to sending. Upon receival, the byte array is first converted back to a JSON string and unmarshalled back to its Java object representation for further processing. Also note that a fixed message size of 353 bytes was used.

Test routines were implemented to send the ping-pong messages between the processes in separate phases called "warmup" and "full-load". The warmup phase was run with 100 iterations and the full-load phase was run with 500 iterations. A result collector object was created and was run on a separate thread to collect the latency results. All the test phases were repeated 200 times to minimize the effects of the cluster network load on the results.

Both the Java socket and MPI latency tests were run separately on a number of compute nodes varying from 1 to 10. They were submitted as batch jobs to the job scheduler of the cluster [8].

OSU Latency Test in Java: In order to complement the work implemented in the previous section, we also wanted to measure the one-way communication latency of the Java sockets and Open-MPI in a simpler way. Therefore, we decided to use the standard OSU Micro-Benchmarks, located in [6]. These tests are offered by the *Ohio State University* to provide ways to measure the network communication performance of MPI configurations. Specifically, the point-to-point *osu_latency* test was chosen that would allow us to benchmark Java socket and Open-MPI latencies in the cluster. However, the OSU tests are implemented in C, therefore to allow for a direct comparison, the *osu_latency* test was also converted to Java.

We implemented three different versions of this test, one using the Open-MPI library and the others using Java Sockets. The Open-MPI version is a one-to-one translation of the test. The first socket test opens and closes a socket each time a message needs to be sent, whereas the second one keeps the connection open until all message transactions are done. The first method provides a better fault tolerant system, where if an endpoint is dead, we would get immediate notification that the socket connection could not be established. On the other hand, the second one provides a performance efficient methodology due to the eliminated overhead of opening and closing a socket when sending a message. Also note that, in all the tests, inter-process communications are done synchronously in a single Java thread.

Finally, the tests were run with 1000 iterations, including the Java Virtual Machine (JVM) warmup period.

4 Experimental Results

4.1 Ping-Pong Latency Test Results

Figure 1(a) shows the average round trip latency (in milliseconds) of a single ping message sent from a single process to N nodes varying in sizes from 1 to 10 when Java's Socket API and Open-MPI library are used on different transmission protocols. Figure 1(b) on the other hand shows the average round trip latency (in milliseconds) of a message sent from one node to the other, indicating the average round trip latency of the point-to-point communication on Java sockets and Open-MPI library. The results show that Java TCP/IP sockets are more stable and efficient than the Java bindings of Open-MPI running on both InfiniBand and TCP/IP stack when node count is less than 8. And the latency values in all the cases converged when the node count is 8 and more. However, what was interesting to observe was that Open-MPI on both InfiniBand and TCP/IP stack showed higher average latency results when the number of nodes are kept between 1 and 3. Since the tests were run multiple times with sufficient number of iterations, the problem seems not to be related with Java's internal mechanisms like failure in optimizing the runtime performance with just-in-time compilation or overhead associated with the garbage collection. Instead, this seems like an internal issue with the Java bindings of Open-MPI.

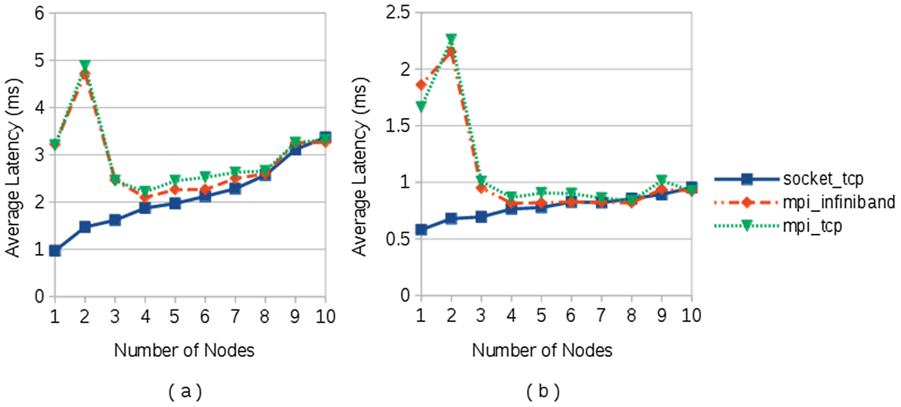


Fig. 1. (a) Average round trip latency (in milliseconds) of a message of size 353 bytes sent from a single node to N nodes. (b) Average round trip latency (in milliseconds) of a point-to-point communication between 2 nodes, using messages of size 353 bytes.

4.2 OSU Point-to-Point Latency Results

Figure 2 shows the average one-way latency (in microseconds) of point-to-point communication of two nodes in varying data sizes. These are the end results of the OSU point-to-point one-way latency benchmarks performed with Java Sockets that use TCP/IP, C implementation and Java bindings of Open-MPI that use InfiniBand and TCP/IP as the underlying communication stack. The Java socket solution included the methodology with opening and closing a socket when a message needs to be delivered each time (java_socket_tcp_open_close) and the one with an always-open connection throughout the test's lifecylce (java_socket_tcp_always_open). It can be seen that the best latency values are obtained with the C implementation of Open-MPI running on InfiniBand interconnect. The slightly higher latency values observed in Open-MPI Java bindings compared to Open-MPI C version seem to have originated from the added overhead of calls being made to the C compiled binaries of Open-MPI. However, the reason for the high latency jump from data of sizes 32768 to 65536 in Open-MPI versions running on TCP interconnect is currently not known. Java sockets on the other hand, performed worse than the provided MPI solutions. Although for small data sizes, the "always open" socket solution performed better than the "open-close" solution, similar results were observed when data size is over 4096 bytes. This means that the added overhead of opening and closing a socket becomes irrelevant when a data with larger size need to be transferred.

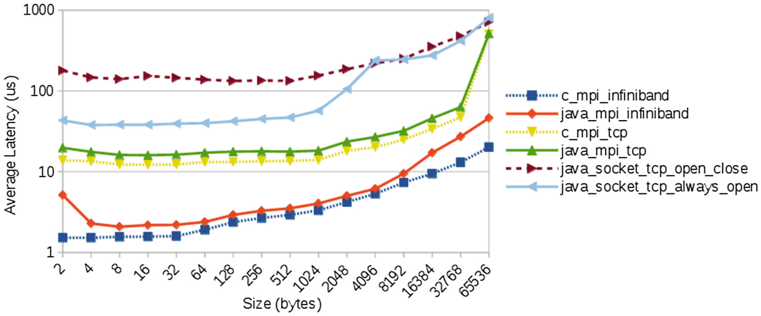


Fig. 2. Average one-way latency (in microseconds) between two nodes on varying data sizes.

The average latency when using OSU tests was lower with a factor of over a thousand than those described in Sect. 4.1. The reason for it is because of the complexity of the project, which was created to serve as a basic simulation of the consensus messaging between the distributed processes. From data marshalling to result collection, a lot of internal processing happens even on a single ping-pong request-response pair, which we believe is expected in such systems.

The other important point was to see that the performance of the socket and the Open-MPI solutions gave opposite results when Figs. 1 and 2 are compared. The main distinction between the two benchmarks is that OSU tests use a synchronized communication strategy with a single Java thread to send and receive messages, whereas the project described above runs on a multi-threaded environment asynchronously. We can conclude that the Java bindings of Open-MPI is not optimized well enough to run concurrently.

5 Conclusion

We have compared the average round trip latency values of the Java bindings of the Open-MPI library running on InfiniBand and TCP/IP stack with the Java's Socket API for TCP/IP communications. The comparison was made with a simple ping-pong test that is intended to reflect a basic distributed database system that uses consensus protocols to reach an agreement on the valid state of the overall system. In addition, we have provided the results collected for the point-to-point osu_latency benchmarks implemented in Java to compare the average latency values between the C implementation and Java bindings of the Open-MPI library and implementations made with the Java Socket API.

We have seen high differences in latencies when osu_latency and ping-pong test results are compared. We concluded that this is caused by the additional functionality that needed to be implemented to give support for a distributed database solution. We have also observed that, regardless of the interconnect, Java bindings of Open-MPI performed poorly than the Java Sockets when multiple Java threads are used to provide concurrent communication.

Although further analysis is needed to investigate the latency performance with a newer version of Java bindings of Open-MPI, these results show that Java Sockets should be preferred instead to develop a distributed database system that will operate in single data centers.

Acknowledgements. Thanks to B.K Muite and A. Jasinski for helpful discussions and suggestions. This work was carried out in the High Performance Computing Center of University of Tartu. This work is partially funded by the Estonian Research Council [IUT34-4].

References

1. Hazelcast the leading in-memory data grid. https://hazelcast.com/. Accessed 06 Apr 2018
2. MPI documents. http://mpi-forum.org/docs/. Accessed 06 Apr 2018
3. mpiJava. http://www.hpjava.org/mpiJava.html. Accessed 06 Apr 2018
4. MPJ Express project. http://mpj-express.org/. Accessed 06 Apr 2018
5. Open source memory-centric distributed database, caching, and processing platform - apache ignite^{TM}. https://ignite.apache.org/index.html. Accessed 06 Apr 2018
6. OSU Microbenchmarks. http://mvapich.cse.ohio-state.edu/benchmarks/. Accessed 07 Apr 2018
7. Rocket cluster - high performance computing center, University of Tartu. https://hpc.ut.ee/en_US/web/guest/rocket-cluster. Accessed 07 Apr 2018
8. SLURM - high performance computing center, University of Tartu. https://hpc.ut.ee/en_US/slurm. Accessed 07 Apr 2018
9. Boysan, M.: mboysan/ping-pong-mpi-tcp: Ping pong test with TCP and MPI. https://github.com/mboysan/ping-pong-mpi-tcp. Accessed 06 Apr 2018
10. Brewer, E.: Towards robust distributed systems. In: PODC, p. 7, January 2000
11. Council, H.A.: Interconnect analysis: 10GigE and InfiniBand in high performance computing. HPC Advisory Council, Technical report (2009)
12. Gray, J., Lamport, L.: Consensus on transaction commit. Technical report, January 2004. https://www.microsoft.com/en-us/research/publication/consensus-on-transaction-commit/
13. Ismail, R., Wati Abdul Hamid, N.A., Othman, M., Latip, R., Sanwani, M.A.: Point-to-point communication on gigabit ethernet and infiniband networks. In: Abd Manaf, A., Sahibuddin, S., Ahmad, R., Mohd Daud, S., El-Qawasmeh, E. (eds.) ICIEIS 2011. CCIS, vol. 254, pp. 369–382. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-25483-3_30

14. Jackson, K., et al.: Performance analysis of high performance computing applications on the Amazon web services cloud. In: 2010 IEEE Second International Conference on Cloud Computing Technology and Science (CloudCom), pp. 159–168. IEEE (2010)
15. Lamport, L.: The part-time parliament, May 1998. https://www.microsoft.com/en-us/research/publication/part-time-parliament/
16. Nowicki, M., Górski, Ł., Grabrczyk, P., Bala, P.: PCJ-Java library for high performance computing in PGAS model. In: 2014 International Conference on High Performance Computing and Simulation (HPCS), pp. 202–209. IEEE (2014)
17. Ongaro, D., Ousterhout, J.: In search of an understandable consensus algorithm. In: Proceedings of the 2014 USENIX Conference on USENIX Annual Technical Conference, USENIX ATC 2014, USENIX Association, Berkeley, CA, USA, pp. 305–320 (2014). http://dl.acm.org/citation.cfm?id=2643634.2643666
18. Skeen, D., Stonebraker, M.: A formal model of crash recovery in a distributed system. IEEE Trans. Softw. Eng. **SE–9**(3), 219–228 (1983). https://doi.org/10.1109/TSE.1983.236608
19. Vega-Gisbert, O., Roman, J., Squyres, J.: Design and implementation of Java bindings in open MPI. Parallel Comput. **59**, 1–20 (2016)
20. Yelick, K., et al.: Titanium: a high-performance Java dialect. Concurr. Comput.: Pract. Exp. **10**(11–13), 825–836 (1998)

Model Based Approach for Testing: Distributed Real-Time Systems Augmented with Online Monitors

Deepak Pal(✉) and Jüri Vain

Department of Software Science, Tallinn University of Technology, Tallinn, Estonia
{deepak.pal,juri.vain}@ttu.ee

Abstract. Testing distributed systems requires an integration of computation, communication and control in the test architecture. This may pose number of issues that may not be suitably addressed by traditional centralized test architectures. In this paper, a distributed test framework for testing distributed real-time systems is presented, where online monitors (executable code as annotations) are integrated to systems to record relevant events. The proposed test architecture is more scalable than centralized architectures in the sense of timing constraints and geographical distribution. By assuming the existence of a coverage correct centralized remote tester, we give a partitioning algorithm of it to produce distributed local testers which enables to meet more flexible performance constraints while preserving the remote tester's functionality. The proposed approach not only preserves the correctness of the centralized tester but also allows to meet stronger timing constraints for solving test controllability and observability issues. The effectiveness of the proposed architecture is demonstrated by an illustrative example.

Keywords: Model-based testing · Real-time database systems
Distributed systems · Low-latency systems

1 Introduction

Testing and verification are vital activities performed during the design and development phase of a system. The continuous growth of systems complexity and high demand of security and reliability in the distributed real-time systems (DRTS) has made their testing a big challenge. Moreover, majority of testing and verification techniques have been developed for the non-real-time systems and they cannot be applied on real-time systems due to timing constrains and concurrency issues. Testing DRTS may pose a number of challenging issues that can not be suitably addressed by traditional centralized remote testing [1]. Major challenges emerge due to severe timing constraints, the tests have to satisfy when the required reaction time of the tester ranges near the message propagation time. These problems restrict the usability of centralized remote testing which

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 142–157, 2018.
https://doi.org/10.1007/978-3-319-97571-9_13

has limited capability of controlling of distributed events, and respecting the timing constraints.

For testing DRTS, designers and developers have frequently used formal verifications techniques during design and development phase of systems [2,3]. In practice, rigorous mathematical proof at the code level is only suitable for small systems due to the state space growth that is exponential in the number of parallel components. Regardless the usage of several state space reduction techniques such as partial order reduction [4] and symbolic model checking [5,6] the problem of scalability still prevents testing and verification of large-scale DRTS. To address this challenge the idea of online monitoring was proposed in [7,8]. In a distributed system the information communicated to different geographical locations (ports) and their time stamps are not globally known. The lack of holistic view makes the coordination of distributed test agents nontrivial. For online monitoring of the distributed systems several authors [8–10] suggested to modify system under test (SUT) to record relevant events (timing and the order of input/output events at different ports) and log the time stamps for global monitoring. The monitored data is collected and integrated to obtain a coherent view of the system. DRTS augmented with online monitors is a prerequisite of distributed model-based testing (MBT) technique presented in this paper. Online monitoring can be performed in-line, in which case the monitors are injected into executable code as annotations [8]. These monitors can be called by applying input to SUT from the location where annotations were placed. However, generating and deploying the monitors for a complex distributed application is a significant engineering effort. Modifying existing distributed systems by instrumenting the annotations may introduce delays and network overhead (probe effect) but there has been lot of research on the implementation of monitors with the aim of obtaining a coherent view of the system and achieving this in a non-invasive manner [8–10].

In this paper, we propose the online monitors based method for testing DRTS, where distributed local testers coordinate test activities via SUT which does not require any external network protocol (required one Δ as proposed in [11]). The main assumption is that monitors are injected in non-invasive manner (without interfering the SUT by introducing timing delay, computation/communication overhead, non-determinism, etc.). The author of [12] has proposed the testing with monitoring systems using status messages and showed such messages can be used to overcome observability and controllability problems for non real-time systems. We give a partitioning algorithm to produce distributed local testers for real-time systems generated by partitioning the given remote tester model. We assume that there already exists a remote tester generated by applying the reactive planning online-tester synthesis method of [13], and its generation is out of scope of this paper. The proposed approach not only preserves the correctness of the test runs defined in the remote tester but also satisfies the timing constraints for solving controllability and observability issues which might be violated in the centralized testing solution. Further, we sketch first experimental results about our implementations, and describe how it can be used to test real-time distributed applications such as real-time databases.

2 Model Based Distributed Testing

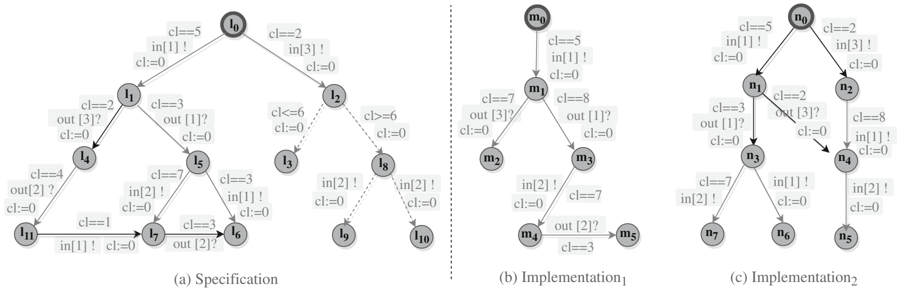
We consider a DRTS, where a system has to respect timing constraints posed on input/output events. These challenges restrict the capability of centralized remote testing which cannot guarantee the controllability of distributed events, and respect their timing constraints because of message propagation time between the tester and SUT. Another aspect to be considered in DRTS is that reaching sufficient test coverage by integration testing of such systems in the presence of numerous latency factors and their interdependency, is out of the reach of off-line testing. Since, off-line testing of such systems is not possible due to the non-deterministic nature of SUT off-line testing approaches need to be replaced by on-line distributed testing.

The need for automated online test generation and tests correctness assurance have given rise to the use of MBT. We interpret MBT in the standard way, i.e. as input/output conformance (IOCO) testing that compares the expected behaviors described by the system model with the observed behaviors of an actual implementation. Due to inherent non-determinism of distributed systems the natural choice is online MBT where the test model is executed in lock step with the SUT. The communication between the model and the SUT involves controllable inputs of the SUT and observable outputs of the SUT which are required for detecting IOCO violations. For detailed overview of MBT and related tools we refer to [14].

2.1 Modelling Implementations of Real-Time Systems

To define our testing architecture formally we need to introduce a semantic foundation for modelling the real-time systems. We describe the notions of timed input output automata (TIOA) introduced by [15,16] as a formalism to model the behavior of real-time systems over time. We consider as time domain $\mathbb{T}$ the set $\mathbb{R}_{\geq 0}$ of non-negative reals called clocks (delays) and Σ as a finite set of actions. For the formal syntax and semantics of TIOA we refer the reader to [16].

Example 1: Consider the TIOA specification $\mathcal{S}pec$ shown in Fig. 1(a), $in[1]!$, $in[2]!$, $in[3]!$ denotes the input to the system and $out[1]?$, $out[2]?$, $out[3]?$ denotes the output produced in response to input to the system. The timed $\mathcal{S}pec$ can be expressed in similar language as follow: exactly at 5 time units after the system received the input $in[1]!$ and produces either output $out[3]?$ exactly at 2 time units or, failing to do that, output $out[1]?$ exactly at 3 time units. The clock cl is set to 0 just after passing the transition. A timed trace ρ is a sequence of timed-stamp actions followed with a delay, $\rho_{\mathcal{S}eq(\mathcal{R})} = (5 \cdot in[1]!)\cdot$ $(8 \cdot out[1]?)\cdot$ $(15 \cdot in[2]!)\cdot$ $(18 \cdot out[2]?) \cdot 0$. We have $\mathcal{S}pec$ After $(5 \cdot in[1]!) \cdot 0 = \{(l_1, 0)\}$, $\mathcal{S}pec$ After $(5 \cdot in[1]!) \cdot (8 \cdot out[1]?).0 = \{(l_5, 0)\}$ $Out(\mathcal{S}pec$ After $(5 \cdot in[1]!) \cdot (7 \cdot out[1]?).0)$ $= \mathbb{T}$, $Out(\mathcal{S}pec$ After $(5 \cdot in[1]!) \cdot (8 \cdot out[1]?).15) = \{in[2]!\} \cup \mathbb{T}$.



(a) Specification

(b) Implementation$_1$

(c) Implementation$_2$

Fig. 1. Models of TIOA specification and implementations

Uppaal Timed Automata (UTA): In our approach UTA [16,17] are used as a formalism to illustrate TIOA to model SUT behavior. This choice is motivated by the need to test the SUT with timing constraints so that the impact of propagation delays between the SUT and the tester can be taken explicitly into account when the test cases are generated and executed. UPPAAL is based on the definition of timed automata, which is introduced by [15]. For the full formal syntax and semantics of UTA we refer to [16,17].

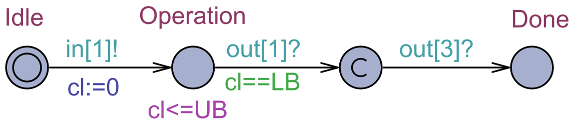


Fig. 2. Modelling pattern of multiport timed automata

Modelling distributed n-Ports: We model a multi-ports TIOA in UTA by splitting the transition with multiple communication actions to a sequence of transitions each labeled with exactly one I/O-action and connected via committed locations, so that all ports of such group are updated instantaneously in the order they are specified in the tuple. In Fig. 2 the labels on the transition represent the i/o actions and the transition tuple $(l_0,\, l',\, in[1]!\ /(out[1]?,\, out[3]?))$ is represented by sequence of transitions each labeled with exactly one action and connected via committed locations, l_0 represents the *idle*, and l' represents the *Done* location. Let P_{l_n} denotes a set of ports accessible in the physical location l_n where $n \in \mathbb{N}$; I is a n-tuple $(I_1, I_2, \ldots, I_n)$, where I_i is the finite set of inputs at port i, $I_i \cap I_j = \phi$ for $i \neq j$ and i, $j = 1, \ldots n \in \mathbb{N}$. Similarly, O is a n-tuple $(O_1, O_2, \ldots O_n)$, where O_i is finite set of outputs at port i, $O_i \cap O_j = \phi$ for $i \neq j$ and i, $j = 1, \ldots n \in \mathbb{N}$. Each port may receive outputs of other port, i.e. $O = (O_1 \cup \{\varepsilon\}) \times (O_2 \cup \{\varepsilon\}) \times \ldots \times (O_n \cup \{\varepsilon\})$, here $\{\varepsilon\}$ denotes the empty output in response to input to $\mathcal{S}UT$.

2.2 Timed Input/Output Conformance Relation

In order to define the conformance relation, we recall the timed input/output conformance relation (`tioco`) introduced in [18,19]. They propose extension of `ioco` relation with timing constraints including clock valuations with the set of observable actions. The communication between the specification and the SUT involves controllable inputs of the SUT and observable outputs of the SUT. In this work, we introduce the `ioco` at first as defined by [20]. The behaviour of `ioco`-correct implementation should respect after some observations following restrictions: *(i)* the outputs produced by SUT should be the same as allowed in the requirements model;*(ii)* if a quiescent state (a situation where the system cannot evolve without an input from the environment) is reached in SUT, this should also be the case in the model; *(iii)* any time an input is possible in the model, this should also be the case in the SUT. In addition to `ioco`, `tioco` introduces the time delays observable on test interface. This is explained by means of following example (for detailed definition of (`tioco`) we refer to [18,19]. *Example 2:* Consider the timed I/O automata specification $\mathcal{S}pec$ and implementations $\mathcal{I}mpl_1$, $\mathcal{I}mpl_2$ shown in Fig. 1. Based on `tioco` relation, we can verify that if $\mathcal{I}mpl_1$ conforms to $\mathcal{S}pec$, for example: $Out(\mathcal{S}pec \text{ After } (5 \cdot in[1]!)) = \mathbb{T}$ and $Out(\mathcal{I}mpl_1 \text{ After } (5 \cdot in[1]!)) = \mathbb{T}$; $Out(\mathcal{S}pec \text{ After } (5 \cdot in[1]!) \cdot 8) = \{out[1]?\} \cup \mathbb{T}$ and $Out(\mathcal{I}mpl_1 \text{ After } (5 \cdot in[1]!) \cdot 8) = \{out[1]?\} \cup \mathbb{T}$; $Out(\mathcal{S}pec \text{ After } (5 \cdot in[1]!) \cdot 7) = \{out[3]?\} \cup \mathbb{T}$ and $Out(\mathcal{I}mpl_1 \text{ After } (5 \cdot in[1]!) \cdot 7) = \{out[3]?\} \cup \mathbb{T}$, proves that $\mathcal{I}mpl_1$ `tioco` $\mathcal{S}pec$. Similarly, we can prove that $\mathcal{I}mpl_2$ ~~`tioco`~~ $\mathcal{S}pec$ i.e. $Out(\mathcal{S}pec \text{ After } (5 \cdot in[1]!) \cdot (7 \cdot out[3]?)) = \mathbb{T}$ and $Out(\mathcal{I}mpl_2 \text{ After } (5 \cdot in[1]!) \cdot (7 \cdot out[3]?)) = \{in[2]!\} \cup \mathbb{T}$; $Out(\mathcal{S}pec \text{ After } (5 \cdot in[1]!) \cdot (8 \cdot out[1]?).18) = \{out[2]?\} \cup \mathbb{T}$ and $Out(\mathcal{I}mpl_2 \text{ After } (5 \cdot in[1]!) \cdot (8 \cdot out[1]?).18) = -$.

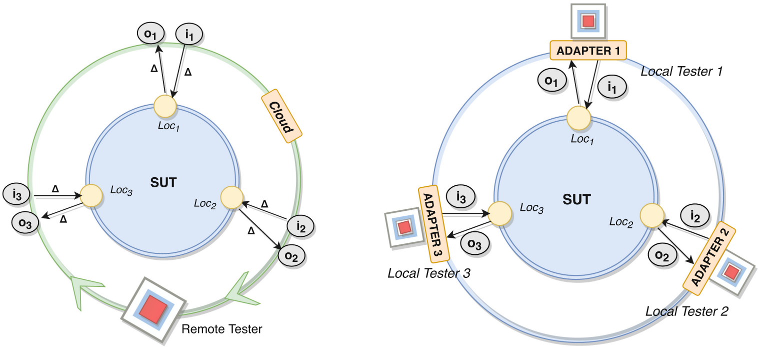
3 Challenges of Centralized Remote Testing

In [1], authors addressed the conformance testing of remote SUTs and proposed the testing architecture which is composed of one FIFO for each direction of communication between SUT and remote tester with communication latency bounded by Δ. Using Δ-testability criteria, it was shown that if the SUT ports are remotely observable and controllable then 2Δ-condition is sufficient for satisfying timing correctness of the test. Here, Δ denotes an upper bound of message propagation delay between tester and SUT ports. Though this approach works reasonably well for systems with sufficient timing margins, it cannot be extended to systems with the timing constraint less than 2 Δ. This means that the actions may not reach the port in time and as a result, the testing becomes infeasible in such systems.

Impact of latency in remote testing: In remote testing the reactions are not always received in the order their stimuli are sent. In order to control the simultaneous test inputs, tester should not wait to receive outputs before sending the next input to $\mathcal{SUT}$. *Example 3*: Consider the timed I/O automata specification $\mathcal{S}pec$ in Fig. 1(a). Assume that the propagation latency exactly 3 time units between $\mathcal{SUT}$ and $\mathcal{S}pec$, which means if $\mathcal{S}pec$ has to apply input to $\mathcal{SUT}$, it should send that input 3 time units earlier, so that $\mathcal{SUT}$ receive the input on time as specified in specification. To maintain the propagation delay, the $\mathcal{SUT}$ and tester shall observe the timed trace shown in Table 1. The $\mathcal{S}pec$ sends second input $in[1]!$ at 9 time units before receiving outputs $out[3]?$, and $out[2]?$ in response to previous input $in[1]!$ at 2 time units to $\mathcal{SUT}$. It seems, outputs $out[3]?$, and $out[2]?$ are generated in response to second input $in[1]!$, though $\mathcal{SUT}$ produces outputs as specified in $\mathcal{S}pec$ and sends $out[3]?$, and $out[2]?$ to $\mathcal{S}pec$ before receiving the second input $in[1]!$. However, the emission of an second input $in[1]!$ depends on the reception of an outputs $out[3]?$, and $out[2]?$, because of latency and to maintain it, tester should not wait to receive outputs before sending the input to $\mathcal{SUT}$. This means in remote testing the propagation latency between $\mathcal{SUT}$ and $\mathcal{S}pec$ may lead to unintended interleaving of input/output actions. This affects the generation of inputs for the $\mathcal{SUT}$ and the observation of outputs that may trigger a wrong test verdict.

Table 1. A time trace observed by SUT and remote tester

$\rho_{\mathcal{SUT}}$:	$(5 \cdot in[1]!) \cdot (7 \cdot out[3]?) \cdot (11 \cdot out[2]?) \cdot (12 \cdot in[1]!) \cdot (15 \cdot out[2]?)$
$\rho_{\mathcal{S}pec}$:	$(2 \cdot in[1]!) \cdot (9 \cdot in[1]!) \cdot (10 \cdot out[3]?) \cdot (14 \cdot out[2]?) \cdot (18 \cdot out[2]?)$



(a) Centralized **(b)** Distributed

Fig. 3. Centralized vs distributed test architecture

Consider the remote testing architecture depicted in Fig. 3(a) and its corresponding UTA model in Fig. 4. The SUT shown in figure has 3 ports (p_1, p_2, p_3) in geographically different places with inputs/outputs $in[1]/out[1]$, $in[2]/out[2]$ and $in[3]/out[3]$ at ports $p1$, p_2 and p_3 respectively. The UTA models defines the expected global behavior of any potential SUT. Each expected global behavior is expressed as the sequence of labels of UTA model edges. This global trace is transformed to the *global test sequence*. Another important aspect that needs to be addressed in remote testing is functional non-determinism of the SUT

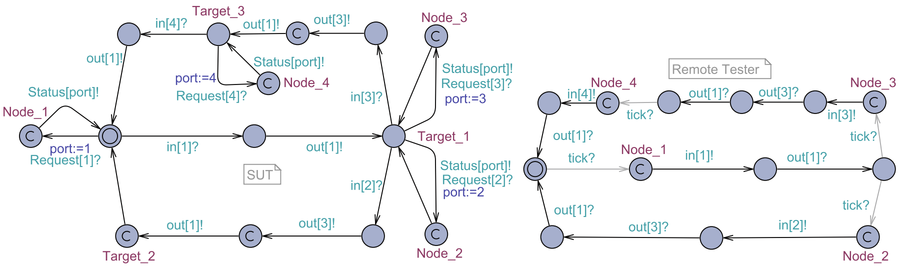


Fig. 4. SUT and remote tester models

behaviour with respect to test inputs. For non-deterministic systems only online testing (generating test stimuli on-the fly) is applicable in contrast to that of deterministic systems where test sequences can be generated offline. The source of timing nondeterminism in remote testing of real-time systems is communication latency between the tester and the SUT that may lead to interleaving of inputs and outputs discussed in *Example 3*. Consequently, the centralized remote testing approach is not suitable for testing a real-time distributed system if the system has strict timing constraints that require reactions faster than 2 Δ. The shortcomings of the centralized remote testing approach are mitigated with overcoming the 2 Δ constraint by partitioning the remote tester into multiple local testers as shown in Fig. 3(b).

4 Distributed Testing

4.1 Distributed Test Architecture

An alternative to the remote testing is distributed testing the architecture of which is shown in Fig. 3(b). Here the remote tester model deployed on a centralized testing architecture is decomposed into a set of communicating (via SUT) local testers, one for each localized interface of the system. Those localized testers communicate with the system by means of adapters whose purpose is both to transfer data between the local interfaces and the localized testers. The approach is supposed to reduce the latency in communication between SUT and local testers (*SUT receiving input and local testers receiving outputs*) caused by two-way messages between the remote tester and SUT. Since the local testers are attached at the same sites as the test ports of SUT the communication delay between a local tester adapter and the SUT port is negligible instead of bidirectional communication needed in case of remote tester.

To implement the model level synchronization of local testers, e.g. to force two test inputs to be inserted at different ports of remote locations at the same time (in the sense of model time), our distributed test execution environment DTRON [21] implements these synchronization channels between local tester models using Spread services [22]. Deterministically controllable test run presumes output observability, which means that the tester attached to some port is waiting for the expected output from this port and after receiving it, propagates it to other testers whose behaviour depends on it. In conventional remote tester case the test stimulus travels from the tester to some SUT port in one Δ and the response from the SUT back to the tester takes another Δ (bidirectional communication), while in the distributed tester's communication with local ports one Δ can be saved.

4.2 Centralized Tester Partitioning Algorithm

We apply Algorithm 1 to transform the centralized remote tester depicted in Fig. 3(a) into a set of communicating distributed local testers, the architecture of which is shown in Fig. 3(b). In this paper we considered a simple case of remote tester model to provide a better theoretical understanding of algorithm. This algorithm covered few cases where tester expects at least one possible output in response to applied input. Line 2–11 adds an adapter model to each local tester instance. The purpose of adding an additional adapter instances to each local tester instance is that it synchronize the local communication between $\mathcal{S}UT$ local ports and a local tester. Its model is derived from remote tester model by adding original channels of $\mathcal{S}UT$ and by renaming channels of local testers. For clarity, notations T_l and A_l represents local tester and local adapter respectively. The channel `tick` denotes the one clock tick (where timing constraints are encoded in `clock Tick Gen` model), each tick represents the real clock variable which track the time elapsed. It provides the time frame for apply input and wait for receiving output from SUT in the same tick. The Channel $\texttt{Request}[l_n]$ represents the special input to SUT that leads to an output $\texttt{status}[\mathit{port}]$ being sent to all SUT-ports and used to coordinate the other local testers via SUT, here argument *port* represents the location from where the $\texttt{Request}[l_n]$ is being applied. The channel $\texttt{status}[\mathit{port}][l_n]$ represents the local communication between adapter and tester. The input $\texttt{in\_}[l_n]$ and output $\texttt{out\_}[l_n]$ are the channels between local adapter and local tester. The ***chan*** $\texttt{in\_}[l_n]\texttt{?}$ represents the reception `R` of input i.e. $T_l \xrightarrow{\texttt{in\_}[l_n]?} A_l$ and ***chan*** $\texttt{out\_}[l_n]\texttt{!}$ represents the emission `E` of output i.e. $A_l \xrightarrow{\texttt{out\_}[l_n]!} T_l$. Similarly, the channels $\texttt{in}[l_n]\texttt{?}$ $\texttt{out}[l_n]!$ are the channels between $\mathcal{S}UT$ and adapter. Now, the construction of local testers, for each locations l_n, we take clone of M^{RT} to be transformed into a location specific local tester instance M^{l_n} (Line 12). The loop in Line 13 says for each clone testers model M^{l_n}, we go through all the edges i/o pair. For clarity, we divided the distribution into two cases, in Line 16, *Case 1* says if the edge has a synchronizing channel i.e. $\texttt{in}[l_n]/\texttt{out}[l_n]$ and the channel belongs to same port location $l_n \in P_{l_n}$ then we add the reception (co-action) of chan $\texttt{status}[\mathit{port}][l_n]$ and *Rename* the ***chan*** $\texttt{in}[l_n]\texttt{!}$, $\texttt{out}[l_n]\texttt{?}$ to $\texttt{in\_}[l_n]\texttt{!}$, $\texttt{out\_}[l_n]\texttt{?}$ as shown in Fig. 5. Basically, idea is to minimize the automata M^{l_n} by removing all synchronizing channels that do not belong to this location. In Line 20, *Case 2* says if input $\texttt{in}[l_j] \wedge$ and output $\texttt{out}[l_j]$ does not belong to same port location $l_j \notin P_{l_i}$ then replace those channels with channel $\texttt{status}[l_j][l_i]$. Similarly, in Line 24, if there is an output $\texttt{out}[l_i]$ generated by SUT in response to input $\texttt{in}[l_j]$, channel $\texttt{status}[l_j][l_i]$ is followed by $\texttt{out}[l_i]$.

Algorithm 1. Automated Construction of Adapter and Local Testers

```
input: $M^{RT}$; output: $\|_n M^{\mathcal{DT}}$ $n \in \mathbb{N}$ ;
for all $l_n \in Loc(\mathcal{SUT})$ do                                    ▷ $n \in \mathbb{N}$
    Add chan tick?                                          ▷ R: $Clock\_Gen \rightarrow A_l$
    Add chan Request[$l_n$]!                                ▷ E: $A_l \rightarrow \mathcal{SUT}$
    Add chan in_[$l_n$]?                                    ▷ R: $T_l \rightarrow A_l$
    Add chan in[$l_n$]!                                     ▷ E: $A_l \rightarrow \mathcal{SUT}$
    Add chan status[port]?                                  ▷ R: $\mathcal{SUT} \rightarrow A_l$
    Add chan status_[port][$l_n$]!                          ▷ E: $A_l \rightarrow T_l$
    Add chan out[$l_n$]?                                    ▷ R: $\mathcal{SUT} \rightarrow A_l$
    Add chan out_[$l_n$]!                                   ▷ E: $A_l \rightarrow T_l$
end for
copy $M^{RT}$ to $M^{l_n}$                                  ▷ take clone at each location
for all $M^{l_n}$, $n \in \mathbb{N}$ do
    for all chan[$l$]: in[$l_n$]/out[$l_n$] pairs $\in M^{l_n}$ do
        Case 1: Consider arbitrary port, n = i $\in l_n$
        if edge.in[$i$] $\wedge$ edge.out[$i$] $\wedge$ $i \in P_{l_i}$ then
            Add chan status_[port][$i$]?                    ▷ R: $A_l \rightarrow T_l$
            Rename chan in[$i$]!, out[$i$]? to in_[$i$]!, out_[$i$]?
        end if
        Case 2: Consider arbitrary port, n = j $\in l_n$
        if edge.in[$j$] $\wedge$ edge.out[$j$] $\wedge$ $j \notin P_{l_i}$ then
            Replace chan in[$j$]!, out[$j$]? to status_[$j$][$i$]?      ▷ R: $A_{l_j} \rightarrow T_{l_i}$
        end if
        if edge.in[$j$] $\wedge$ edge.out[$j$]/out[$i$] where $j \notin P_{l_i}$ $\wedge$ out[$i$] $\in P_{l_i}$ then
            Replace chan in[$j$]!, out[$j$]? to status_[$j$][$i$]? $\rightarrow$ edge.out[$i$]
        end if
    end for
end for
```

M^{RT}: Remote Tester Model; $\|_n M^{\mathcal{DT}}$: Communicating Distributed Testers; $l_n \in Loc(\mathcal{SUT})$: represents the number of ports of SUT; $Clock\_Gen$: timing constraints encoded in model; R, E: represents Reception and Emission of channel; A_l: Adapter Model; T_l: Local Tester Model;

Figure 5 represents the generated parameterized local tester with corresponding parameterized adapter model where L denotes the geographical location.

In remote testing, tester generates an input for the SUT, waits for the result and continues with the next set of inputs and outputs until the test scenario has been finished. Thus, the tester has to wait for the duration it takes the signal to be transmitted from the tester to the SUT's ports and the responses back from ports to the tester. Therefore, $\mathcal{SUT}$ conforms M^{RT} if following constraints are satisfied: *1. Order constraint* In Fig. 4, remote tester can generate inputs $in[2]!$ (Node_2) or $in[3]!$ (Node_3) only after receiving $out[1]?$ (Node_1) in response to applied input $in[1]!$; *2. Timing constraint* In the case of SUT being distributed in a way that signal propagation time is non-negligible, this can lead into a situation where the tester is unable to generate the necessary input for the SUT in time due to message propagation latency. For example, if the inputs $in[2]!$ (Node_2) or $in[3]!$ (Node_3) has clock constraints and input $in[1]!$ (Node_1) executed before

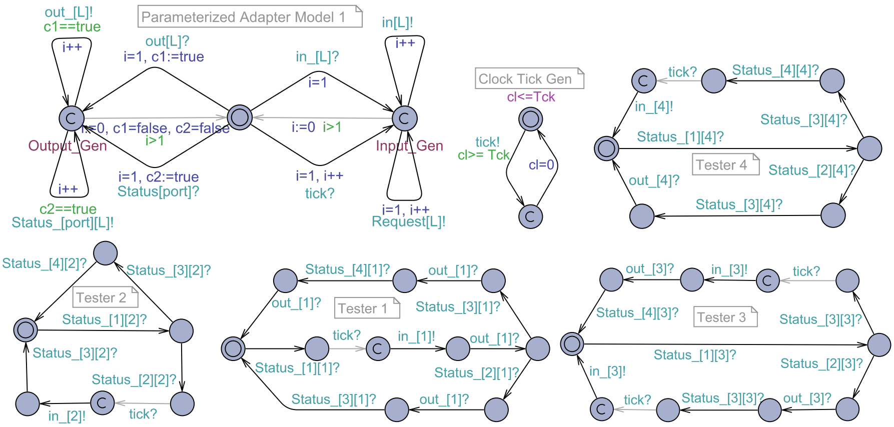


Fig. 5. Parameterized adapter and local tester models

them and waiting to receive output from SUT, then the delay separating two consecutive inputs must satisfy the condition clock constraints at tester and propagation latency $\Delta \leq$ max waiting time by SUT.

Let us consider now, $\mathcal{SUT}$ conforms $\shortparallel_n$ $M^{\mathcal{DT}}$ if order constraints on observable actions and timing constraints are satisfied. To prove both the constraints on actions applied by local testers, we have following test sequence: *(i)* We consider a clock constraint encoded in the tick model, where each action triggered has to finish execution in one tick, where a tick is synchronized with all local tester models. Timings constraints encoded in the model are fictitious and must be respected by real SUT; *(ii)* We assume that tester apply input $\texttt{Request}[l_i]$ (before actual input) to SUT that leads to an output $\texttt{status}[i]$ sent to all SUT-ports in one tick and reset the clocks; *(iii)* Immediately after executing the $\texttt{Request}[l_i]$, the tester sends $in[l_i]!$ to SUT. This may lead to an output at all port with in one tick cycle. Any output actions observed outside the time frame is consider as non-conformance with SUT. We show that problem of controllability and observability in DRTS can be overcome by testing systems with timing constraints, provided monitors (input $\texttt{Request}[l_i]$ and Outputs $\texttt{status}[l_i]$) are implemented correctly.

Observability Problem: The input $\texttt{Request}[l_i]$ applied to SUT results in output $\texttt{status}[l_i]$ to all ports. Output $\texttt{status}[l_i]$ has sufficient information and allows the other local testers to guarantee the order and timing constraints of incoming input/output actions from port l_i. Each local tester recognizes the port where input was applied for which the specific output response was received at their ports, that overcomes the observability problems among local testers.

Controllability Problem: Similarly, if $tester_j$ at location j has to apply input after input $in[l_i]!$ triggered by $tester_i$ at location i. The $tester_j$ has to wait for following actions: For $tester_j$ to apply input upon execution of input from $tester_i$, it wait for reception of output $\texttt{status}[l_i]$ in response to $\texttt{Request}[l_i]$ and output

$out[j]?$ (if any) generated by SUT in response to input $in[i]!$ from location i. As we assumed that communication delay between testers and SUT is negligible, it eliminates the possibility of introducing delay and overlapping messages with others. Also clock constraints encoded in tick model force to respect timing constraints. Hence, waiting for the reception of $\mathtt{status}[l_i]$ and output $out[j]?$ allows the $tester_j$ to overcome problem of controllability.

5 Use Case: MBT for Distributed Real-Time Database System

To illustrate the proposed test architecture and distributed tester interactions, we present an example of testing distributed real-time database system. Figure 6 shows the communication among distributed nodes. *The System Description*: Complex distributed transactions of database often involve parallel execution of its sub-transactions at different nodes where each node has replicas of parts of the distributed database. In each participant node, sub-transactions are managed by a cohort process. A global transaction (a master node in which each global transaction is submitted) is characterized by its arrival time and its deadline while, within a participant node, a sub-transaction is characterized by its arrival time, its execution duration and its deadline. To meet the deadline of a global transaction, all of its parallel sub-transactions have to be finished in time. In comparison to a local transactions (which involves execution at only one node), a global transactions may find it much harder to meet its deadline because it may happen that at least one of its sub-transactions runs into an overloaded node. The communication between master and distributed nodes is shown in Fig. 6. Another problem with complex real-time databases occurs when a global transactions consists of parallel and serial sub-transactions. If one parallel sub-transactions is late, then the whole transactions is late. The problem of assigning deadlines to parallel and serial sub-transactions of complex distributed sub-transactions in a real time system has been studied through simulation where the scheduler must estimate the execution times of the sub-transactions and assign them to processors in such a way that all will finish before the deadline of the global task [23].

Testing of such distributed real-time transactions scheduling protocols is completely out of scope of centralized remote test architecture if the SUT has strict timing constraints that require reactions faster then 2 Δ. Hence to prove the advantages of distributed test architecture over centralized tester and show applicability of distributed testers, we consider a worst case scenario to test real-time database systems under overload situations. Distributed applications, such as web-based services, electronic commerce, mobile telecommunication system, etc., combine active database functionality with critical timing constraints and integrated system monitoring. In order to enhance the performance and the availability of such applications one of the major issues is to develop efficient replica concurrency control protocol that is able to tolerate the overload of the distributed system. Dynamic overloads situations cannot be handled since

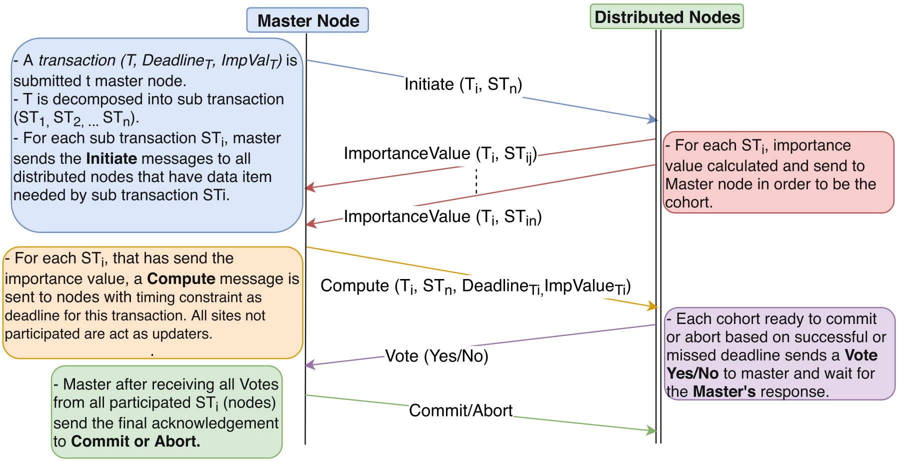

Master Node
Distributed Nodes
- A transaction (T, Deadline$_T$, ImpVal$_T$) is submitted t master node.
- T is decomposed into sub transaction (ST$_1$, ST$_2$, ... ST$_n$).
- For each sub transaction ST$_i$, master sends the **Initiate** messages to all distributed nodes that have data item needed by sub transaction STi.
Initiate (T$_i$, ST$_n$)
ImportanceValue (T$_i$, ST$_{ij}$)
ImportanceValue (T$_i$, ST$_{in}$)
- For each ST$_i$, importance value calculated and send to Master node in order to be the cohort.
- For each ST$_i$, that has send the importance value, a **Compute** message is sent to nodes with timing constraint as deadline for this transaction. All sites not participated are act as updaters.
Compute (T$_i$, ST$_n$, Deadline$_{Ti}$, ImpValue$_{Ti}$)
- Each cohort ready to commit or abort based on successful or missed deadline sends a **Vote Yes/No** to master and wait for the **Master's** response.
Vote (Yes/No)
- Master after receiving all Votes from all participated ST$_i$ (nodes) send the final acknowledgement to **Commit or Abort.**
Commit/Abort

Fig. 6. Communication messages exchanged among distributed nodes

the system is designed to handle the worst case. In fact, if the system is not designed to handle overloads, the effects can be catastrophic and some primordial transactions of the application can miss their deadlines. However, several dynamic preemptive transaction scheduling with integrated system monitoring have been adopted to address overload situations with contingency plans but their correctness would necessitate a rigorous approach to testing.

Use Case: Testing Replicated Real-Time Databases in an Overloaded Mode: Based on general scheduling protocol, communication of which is shown in Fig. 6, building overload situation on distributed nodes by sending multiple test inputs (triggers sub-transactions) is not possible with centralized remote tester. Essentially, for parallel and serial sub-transactions, computation in the master node cannot start until the information from distributed cohort nodes with the longest delay has arrived. In addition to communication delay among distributed nodes, remote tester impose 2 Δ delay challenge. Therefore, generating multiple test inputs (triggers sub-transactions) simultaneously at distributed nodes in order to create overload situations with remote tester is out of the scope of centralizes test architecture. Using distributed test architecture, we can deploy the communicating local testers at distributed nodes. Therefore, each local tester can coordinate each other via SUT and can generate dynamic overload (the worst case) where system can be tested against overloading of real-time transactions with hard, firm or soft deadlines. *The System Model*: The corresponding local distributed tester models are depicted in Fig. 7. Each distributed node can act both as master and participant node which helps to generate dynamic overload on multiple nodes simultaneously. As discussed in Sect. 4, generated distributed testers are more scalable and efficient than centralized architectures in the sense of timing constraints and geographical distribution which enables testing a real-time database system under dynamic overload situations.

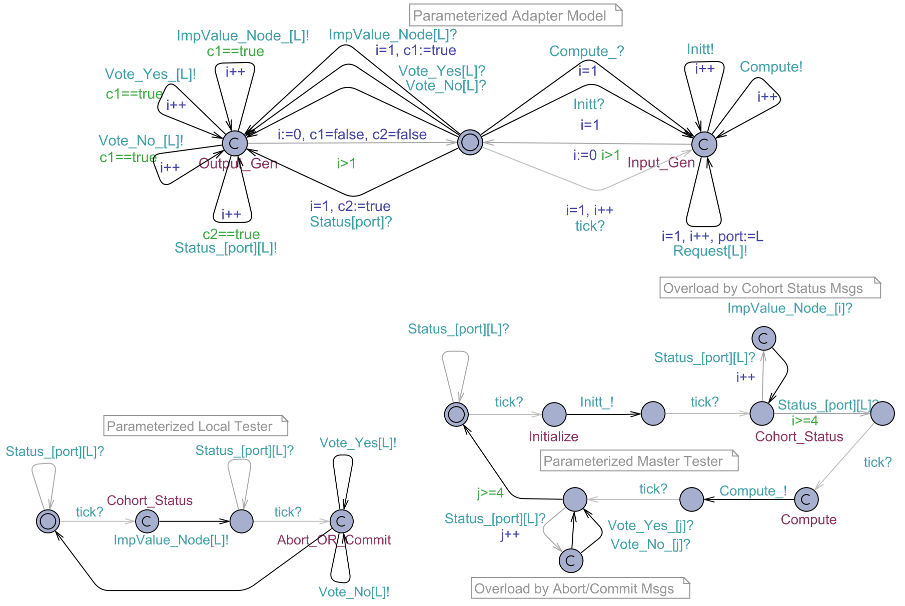


Fig. 7. Parameterized adapter and local tester models

6 Related Work

As for broader context of distributed testing the early works focused on testing distributed non real-time systems [13,24,25]. The theory of testing distributed real-time systems has gained interest only in recent years when global time keeping techniques emerged. For instance the authors of [26] assumed that each local tester has a local clock which adds timestamps to its observations. It is assumed that the proposed approach provides additional information regarding the causality between actions performed at different ports. But the approach relies on strong assumption that it is known how much local clocks can differ between synchronization events which appears to be unrealistic. Though approach works reasonably well for distributed testing it cannot be used if specifications contains strict timing requirements, especially if there are requirements regarding the relative timing of actions at different ports. Pioneering results on testing timing correctness with remote testers was proposed in [1] where a remote abstract tester was proposed for testing distributed systems in a centralized manner. It was shown that if the SUT ports are remotely observable and controllable then 2Δ-condition is sufficient for satisfying timing correctness of the test. Here, Δ denotes an upper bound of message propagation delay between tester and SUT ports. Though this approach works well for systems with sufficient timing margins, it cannot be extended to systems with the timing constraint more strict than 2 Δ. This means that the test inputs may not reach the input port in time and as a result, the testing becomes infeasible in such systems. The shortcomings of the centralized remote testing approach are mitigated by partitioning the remote tester into multiple local testers that are deployed in the same locations with the SUT component they are testing [11] where the controllability and observability problems are resolved by allowing the local testers to exchange messages through external reliable communication independent of the SUT.

7 Conclusion

In this paper, a distributed test framework for testing of a DRTS augmented with monitors is presented, where online monitors are used to record relevant events (timing and order of input/output events at test interface ports). Online monitored data is used to obtain a coherent view of the system and to simplify distributed testing, where local testers synchronize with each other via communicating with these monitor. The proposed test architecture is test reaction time wise more scalable than centralized remote test architecture for testing large number of geographical locations (ports) in a system. We give a partitioning algorithm to produce automated distributed local testers from given remote tester model. The proposed approach not only preserves the functional correctness of the centralized remote testers but also satisfy stronger timing constraints needed for solving distributed test controllability and observability issues.

In spite of the advantages of proposed approach, a key requirement for real-time distributed system online monitors is low overhead. The main assumption is that monitors are injected in non-invasive manner (without interfering the SUT by introducing timing delays, computation/communication overhead, especially when tasks run in parallel, they can introduce non-determinism, etc.). As a near future work, we plan to implement the proposed test architecture using MBT platform DTRON as an test execution platform used for facilitating distributed testers deployment and management. Moreover, we plan to implement generic online monitors in non-invasive manner using Java which allows the basic assertions to be inserted into the system.

References

1. David, A., Larsen, K.G., Mikučionis, M., Nguena Timo, O.L., Rollet, A.: Remote testing of timed specifications. In: Yenigün, H., Yilmaz, C., Ulrich, A. (eds.) ICTSS 2013. LNCS, vol. 8254, pp. 65–81. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41707-8_5
2. Wachter, B., Genon, A., Massart, T., Meuter, C.: The formal design of distributed controllers with dSL and spin. Formal Aspects Comput. **17**, 177–200 (2005)
3. Cimatti, A., et al.: NuSMV 2: an opensource tool for symbolic model checking. In: Brinksma, E., Larsen, K.G. (eds.) CAV 2002. LNCS, vol. 2404, pp. 359–364. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45657-0_29
4. Godefroid, P. (ed.): Partial-Order Methods for the Verification of Concurrent Systems - An Approach to the State-Explosion Problem. LNCS, vol. 1032. Springer, Heidelberg (1996). https://doi.org/10.1007/3-540-60761-7

5. Clarke, E., Grumberg, O., Peled, D.: Model Checking. The MIT Press, Cambridge (1999)
6. McMillan, K.L.: Symbolic model checking: an approach to the state explosion problem. Carnegie Mellon University (1992)
7. Leucker, M., Schallhart, C.: A brief account of runtime verification. J. Log. Algebr. Program. **78**(5), 293–303 (2008)
8. Goodloe, A., Pike, L.: Monitoring distributed real-time systems: a survey and future directions (NASA/CR-2010-216724). In: Havelund, K., Rosu, G. (eds.) Synthesizing monitors for safety properties (2010)
9. Bauer, A., Leucker, M., Schallhart, C.: Model-based runtime analysis of distributed reactive systems. In: Australian Software Engineering Conference, p. 10 (2006)
10. Sen, K., Vardhan, A., Agha, G., Rosu, G.: Efficient decentralized monitoring of safety in distributed systems. In: Proceedings of 26th International Conference on Software Engineering, pp. 418–427 (2004)
11. Vain, J., Halling, E., Kanter, G., Anier, A., Pal, D.: Model-based testing of real-time distributed systems. In: Arnicans, G., Arnicane, V., Borzovs, J., Niedrite, L. (eds.) DB&IS 2016. CCIS, vol. 615. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40180-5_19
12. Hierons, R.M.: Using status messages in the distributed test architecture. Inf. Softw. Technol. **51**(7), 1123–1130 (2009)
13. Vain, J., Kaaramees, M., Markvardt, M.: Online testing of nondeterministic systems with reactive planning tester. In: Dependability and Computer Engineering: Concepts for Software-Intensive Systems, pp. 113–150. Hershey (2012)
14. Utting, M., Pretschner, A., Legeard, B.: A taxonomy of model-based testing. Softw. Test. Verif. Reliab. **22**(5), 297–312 (2012)
15. Alur, R., Dill, D.: A theory of timed automata. Theor. Comput. Sci. **126**, 183–235 (1994)
16. Bengtsson, J., Yi, W.: Timed automata: semantics, algorithms and tools. In: Desel, J., Reisig, W., Rozenberg, G. (eds.) ACPN 2003. LNCS, vol. 3098, pp. 87–124. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-27755-2_3
17. Behrmann, G., David, A., Larsen, K.G.: A tutorial on Uppaal. In: Bernardo, M., Corradini, F. (eds.) SFM-RT 2004. LNCS, vol. 3185, pp. 200–236. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-30080-9_7
18. Krichen, M., Tripakis, S.: Black-box conformance testing for real-time systems. In: Graf, S., Mounier, L. (eds.) SPIN 2004. LNCS, vol. 2989, pp. 109–126. Springer, Heidelberg (2004). https://doi.org/10.1007/978-3-540-24732-6_8
19. Mikucionis, M., Larsen, K.G., Nielsen, B.: T-uppaal: online model-based testing of realtime systems. In: 19th IEEE International Conference on Automated Software Engineering, pp. 396–397. IEEE Computer Society (2004)
20. Tretmans, J.: Test generation with inputs, outputs and repetitive quiescence. Softw. Concepts Tools **17**(3), 103–120 (1996)
21. Anier, A., Vain, J., Tsiopoulos, L.: DTRON: a tool for distributed model-based testing of time critical applications. Est. Acad. Sci. **66**, 75–88 (2017)
22. The spread toolkit. http://spread.org/. Accessed 24 Nov 2016
23. Saad-Bouzefrane, S., Kaiser, C.: How to manage replicated real-time databases in an overloaded distributed system. In: Real-Time Systems Conference (2003)
24. Cacciari, L., Rafiq, O.: Controllability and observability in distributed testing. Inf. Softw. Technol. **41**, 767–780 (1999)

25. Ahmed, K.: A temporal approach for testing distributed systems. IEEE Trans. Softw. Eng. **28**, 1085–1103 (2002)
26. Hierons, M., Merayo, G., Núñez, M.: Timed implementation relations for the distributed test architecture. Distrib. Comput. **27**, 181–201 (2014)

Knowledge and Ontologies

Domain Ontology for Expressing Knowledge of Variants of Thermally Modified Wood Products

Hele-Mai Haav(✉) and Riina Maigre

Department of Software Science, Tallinn University of Technology, Akadeemia tee 15a,
12618 Tallinn, Estonia
helemai@cs.ioc.ee, riina@ioc.ee

Abstract. The thermally modified wood producer Thermory AS manufactures about 400 different products, which are ordered in large number of variants that makes the expression of the product variant knowledge and its validation very important. In this paper, we express knowledge of product variants as domain ontology in order to capture the product knowledge in the way that is consistent and shareable between humans and machines. Using Ontology Web Language (OWL) as Description Logics (DL) based ontology representation language enables to detect inconsistency in the product knowledge and customer order requirements. Constraints on valid product variants are expressed as OWL class expressions and as rules in Semantic Web Rule Language (SWRL). The provided knowledge representation method makes it possible to reduce combinatorial complexity of description of product variants and to place correct manufacturing orders saving time and money for the company.

Keywords: Ontology · OWL · Product variant management · SWRL
SPARQL

1 Introduction

Today many businesses need to deliver products that have variations in some attribute (or parameter) values. A certain combination of these attribute values on a particular product is called a variant or variation. In wood industry, wood products can have a set of common parameters for some product categories and in addition several variations in values of some other parameters. Customer orders specify values of variant parameters of an ordered product.

In traditional Enterprise Resource Planning (ERP) or Material Resource Planning (MRP) systems product variations are managed using Bill of Materials (BOM) with parameters[1] (or variant/matrix BOM). Product variant management is related to more general problem of product configuration. Several product configuration systems are available as parts of ERP (e.g. SAP[2]) or as standalone systems (e.g. Productoo[3]).

[1] www.mrpeasy.com.

[2] www.sap.com.

[3] https://www.web4industry.com/product-configurator/.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 161–171, 2018.
https://doi.org/10.1007/978-3-319-97571-9_14

All these traditional product or variant configuration systems represent product variant knowledge in the form of database tables (or matrices) creating, if necessary, a table including huge number of variants of the same product (i.e. for all combinations of values of parameters). This kind of knowledge representation is not well reusable, manageable, shareable, and interoperable with other systems used in a global enterprise. Therefore, one of the most important challenges of solving the product configuration as well as the product variant management problem is related to the knowledge representation that requires the expressive language for describing product variations and their constraints as well as customer preferences.

Ontology languages like OWL [9] provide the expressive and explicit way of capturing domain knowledge as well as reasoning on the basis of the described knowledge. They also enable reusability of the represented knowledge in other systems and semantic interoperability between different possibly distributed systems.

The thermally modified wood producer Thermory AS is a SME that currently does not use any well-known ERP systems but relies on its in-house developed information system (IS). We chose an ontology engineering approach for solving the product variant management problem for Thermory. To do that, we first created ontology based representation of available thermally modified product variations and then used it for resolving inconsistencies in the product variant knowledge as well as to detect inconsistency between the product variant knowledge and customer requirements. Ontology is represented in OWL and constraints on valid product variants are expressed as class expressions in OWL and as rules in Semantic Web Rule Language (SWRL) [4]. The DL reasoner Pellet [11] is used for ontology reasoning as it well supports SWRL rules. The novelty of our approach comparing to the traditional variant BOM lies in the significant reduction of number of product variant combinations to be described and managed as well as in the possibility to use DL reasoning services.

The paper is structured as follows. In Sect. 2 we give a background of the problem and in Sect. 3 we consider some related works. Section 4 is devoted to our original approach of expressing knowledge of product variants using OWL ontologies and SWRL rules. Conclusion and future work are presented in Sect. 5.

2 Motivation and Background

The thermally modified wood producer Thermory AS[4] is an Estonian company specializing in thermally modified solid wood flooring, decking, cladding and sauna products. The Thermory brand has become well-established in the United States and Canada, and has been shipped to over 55 countries around the world. Thermory AS uses chemical-free thermal modification process, where properties of wood are altered using only heat and steam[5]. As a result of the thermal modification process, wood's durability and resistance to mold and rotting increases. Due to its properties, the main wood species used by Thermory AS is ash, but pine, pecan, hickory and birch are also used. Production in Thermory's factory usually starts with unprocessed saw-timber and includes multiple

[4] http://thermory.com/en/kontakt/about-company.

[5] https://www.thermoryusa.com/modification.

production stages. Main production stages are dehumidification in drying kilns, thermal modification in the thermo-kilns, planing to dimensions and length and planing to profile. In addition to these stages, boards can go through brushing, end-matching and finishing stages.

Depending on the customer order, products can have different thermal modification levels. Two thermal modification levels are used by Thermory AS: medium (peak temperature 190°) and intense (peak temperature 215°). Possible thermal modification levels depend mainly on wood species. Boards have additional variations in dimensions, length, profile and suitable clips. Available profiles and suitable clips depend on dimensions of the board. Thermory's current IS is not able to automatically allocate materials and schedule resources. Therefore, all planning and scheduling is done manually, which is time consuming and costly. We have been working with Thermory in order to develop industry and production specific algorithms for automating the planning of materials and resources in the factory. As a part of this project a need for the list of descriptions of all product variants arose. Such knowledge does not currently exist in Thermory's IS, but it is necessary for automating the production planning.

3 Related Works

Our approach is indirectly related to works on ontology based product configuration. One of the first works (published in late 90s) that uses DL based knowledge bases for product configuration is [8]. They have built configurators based on DL based knowledge representation system CLASSIC [10] for a number of large telecommunications products sold by AT&T and Lucent Technologies.

At the same time the work towards a general ontology of configuration was developed in order to reuse and share configuration knowledge [12]. This ontology includes concepts like components, attributes, resources, ports, contexts, functions, constraints, and relations between these. It is formalized in Ontolingua [3] based on KIF [2] that lacks reasoning mechanism for checking the consistency of a knowledge base that is available in DL based languages.

In [15] an ontology-based product configuration model was developed and formalized using OWL and SWRL. A similar approach can be found in [14], where focus is on the semantics of constraints of product configuration that cannot be expressed by OWL. They provide a rule based ontological formalism for describing product structure and constraints of a product configuration and checking its validity.

Interesting relationships can be found between feature oriented domain analysis (FODA) [1] used basically for software line production and our ontology based method suggested for product variant management in manufacturing. The authors of [5] analyzed similarities and differences of feature models of FODA [1] and ontology based domain analysis methods. According to their work, similarities include using a concept vocabulary, enabling the expression of property and class hierarchies, and providing a constraint definition capability. In FODA, the latter is used for variability reduction but in ontology based domain analysis constraints are used for the description of property restrictions in class expressions. Both analysis methods allow to describe semantics of a domain and can be represented in machine readable form. Therefore, the Authors of [1] conclude that ontologies could effectively replace FODA models. As the advantages, ontology based analysis provides more expressive language than FODA and includes additional capabilities like reasoning and querying (via DL or SPARQL query support).

To the best of our knowledge, the only work that is tightly related to ontology based product variant management in manufacturing domain is devoted to the creation of the product feature ontology in [7]. This is intended to the management of the feature-based product line engineering in very large and complex product line organizations. Their goal is similar to what we have, to create multilevel ontology in order to significantly reduce number of product feature combinations to be managed comparing to using feature matrices. However, the goal and the scope (automotive industry) of their ontology are different from our ontology and there is no information about formalization of this ontology in any formal ontology language.

4 The Approach to Expressing Knowledge of Variants of Thermally Modified Wood Products

4.1 An Example

In a running example we consider a set of thermally modified wood product families like ash decking and cladding boards, pine decking and cladding boards and spruce decking boards that form the largest share of the production of Thermory.

Each product family includes a number of different products (i.e. parent products) that can have variants according to the values of some parameters (see Fig. 1).

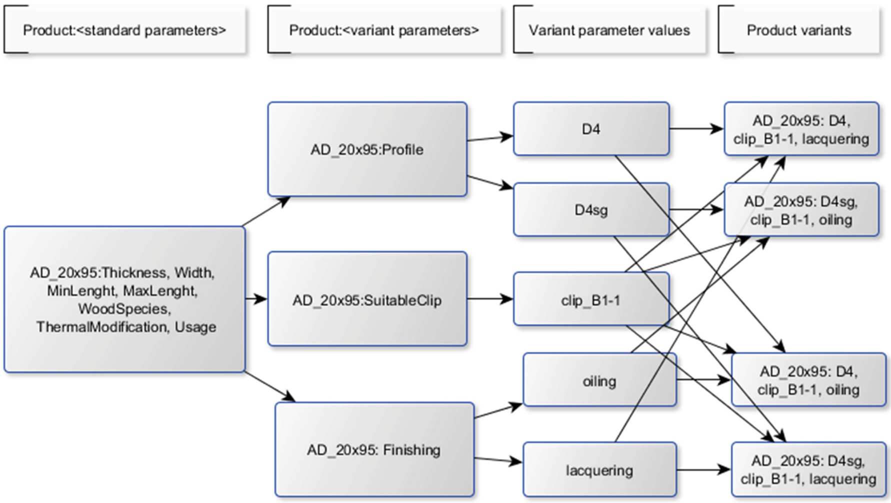
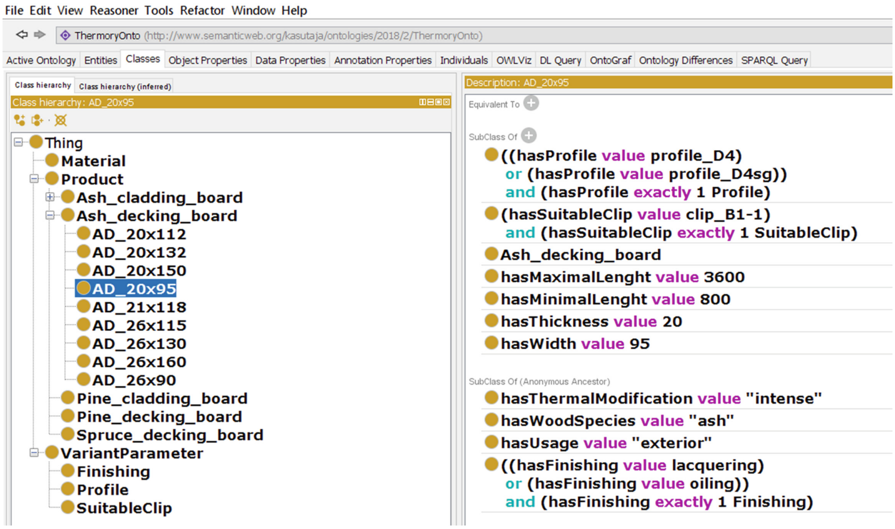

Product:<standard parameters>
Product:<variant parameters>
Variant parameter values
Product variants
AD_20x95:Profile
D4
AD_20x95: D4, clip_B1-1, lacquering
D4sg
AD_20x95: D4sg, clip_B1-1, oiling
AD_20x95:Thickness, Width, MinLenght, MaxLenght, WoodSpecies, ThermalModification, Usage
AD_20x95:SuitableClip
clip_B1-1
oiling
AD_20x95: D4, clip_B1-1, oiling
AD_20x95: Finishing
lacquering
AD_20x95: D4sg, clip_B1-1, lacquering

Fig. 1. An example of product variants in Thermory AS

Product families and product hierarchy are illustrated by ontology class hierarchy in Fig. 2.


File Edit View Reasoner Tools Refactor Window Help
ThermoryOnto (http://www.semanticweb.org/kasutaja/ontologies/2018/2/ThermoryOnto)
Active Ontology
Entities
Classes
Object Properties
Data Properties
Annotation Properties
Individuals
OWLViz
DL Query
OntoGraf
Ontology Differences
SPARQL Query
Class hierarchy
Class hierarchy (inferred)
Class hierarchy: AD_20x95
Description: AD_20x95
Equivalent To
SubClass Of
SubClass Of (Anonymous Ancestor)

- Thing
  - Material
  - Product
    - Ash_cladding_board
    - Ash_decking_board
      - AD_20x112
      - AD_20x132
      - AD_20x150
      - AD_20x95
      - AD_21x118
      - AD_26x115
      - AD_26x130
      - AD_26x160
      - AD_26x90
    - Pine_cladding_board
    - Pine_decking_board
    - Spruce_decking_board
  - VariantParameter
    - Finishing
    - Profile
    - SuitableClip

SubClass Of

```
((hasProfile value profile_D4)
 or (hasProfile value profile_D4sg))
 and (hasProfile exactly 1 Profile)
(hasSuitableClip value clip_B1-1)
 and (hasSuitableClip exactly 1 SuitableClip)
Ash_decking_board
hasMaximalLenght value 3600
hasMinimalLenght value 800
hasThickness value 20
hasWidth value 95
```

SubClass Of (Anonymous Ancestor)

```
hasThermalModification value "intense"
hasWoodSpecies value "ash"
hasUsage value "exterior"
((hasFinishing value lacquering)
 or (hasFinishing value oiling))
 and (hasFinishing exactly 1 Finishing)
```

Fig. 2. An excerpt of the product ontology class hierarchy and examples of class expressions

Product variants are made up of a parent product that sets up values for standard parameters and a set of child products that represent different variations of this parent product according the values of variant parameters (see Fig. 1).

For example, standard and variant parameters of the ash decking board AD_20x95 are presented in Fig. 1. Some of standard parameters are common for the whole product family (e.g. ThermalModification, Woodspecies, Usage) and some are not (e.g. Thickness, Width, etc.). There are shown three variant parameters in Fig. 1 as follows: profile, suitable clip and finishing. The latter is variant parameter for the whole product family. The given possible parameter values allow creating 4 different product variants. There are more variant parameters. For example, actual length of an ordered product is a variant parameter too. It has associated constraints expressing that its value should be between minimum and maximum length given as product parameters. For some products, actual length parameter can obtain a value only from a specified set of valid values.

4.2 Principles of Ontological Modeling of Knowledge of Product Variants

We developed principles of capturing knowledge embedded in data collected about product families, their parameters, product variants and customer requirements to OWL ontology. The following general guidelines have been worked out:

1. Terminological knowledge about product families and product variants is expressed as the product ontology class hierarchy, object property and data property definitions and class expressions (TBox in DL). For each product family a subclass of the class

Product is defined as a complex class in OWL (see Fig. 2). Knowledge about product variants is defined in subclasses of the corresponding product family class. All subclasses of classes are disjoint. Object and data properties express either standard or variant parameters of a product. They are associated with the class Product having it as a domain.

2. Individuals (ABox in DL) are used to represent value choices of variant parameters (e.g. profile D4). Other parameter values are represented as data property values. In addition, product descriptions that are specified in orders are represented as distinct individuals of a class of a certain product with provided values for variant parameters. They are used in the reasoning process of checking consistency of ontology itself during its design time and for the verification whether specification of an ordered product is in correspondence with valid values of product variant parameters defined in ontology.
3. Constraints are represented as property restrictions in complex class expressions in OWL or as rules in SWRL. Class expressions define a set of individuals belonging to the class. SWRL rules are used to express constraints that cannot be represented by OWL. For example, if some calculations or comparisons are to be performed on product parameter values, then SWRL rules are used.
4. Reasoning is used for the verification of the validity of product variant parameters of an ordered product and for inferring product standard parameters according to the given product hierarchy. The verification is done by using standard ontology consistency check and evaluation of SWRL rules. Some of the rules assert new values to data properties of individuals. DL reasoners use Open World Assumption (OWA) for reasoning meaning that the model may be incomplete and new knowledge may be added that necessarily is not false. This is good for checking partially defined product variants but it creates problems for checking completeness of an individual product description (an individual that corresponds to an ordered product in Abox). For ensuring that a product description in Abox includes all necessary object property and data property assertions that model variant parameters and their values we propose to use SPARQL [13] queries (see Sect. 4.5) to retrieve individuals that do not include necessary properties. After corrections in Abox, if needed, the verification of the validity of the description of the given individual product in ABox can be performed by a DL reasoner.
5. During the evolution of ontology (according to the evolution of product variants) its consistency needs to be checked again by a DL reasoner before it can be used for the validation of parameters of an individual ordered product.

4.3 Definition of Product Ontology Classes

The product ontology class hierarchy corresponding to our running example is presented in Fig. 2. The class hierarchy contains disjoint classes for product families, product variants and variant parameters. A product family class is defined as a complex class with class expression including data property value restrictions for standard parameters of the product family products.

In addition, according to the specific product family, this class expression may define common property restrictions over object properties corresponding to variant parameters of a product family. Product variants are defined as subclasses of a product family class and their class descriptions specify only data property values and object property restrictions that correspond to the specific variant parameters of this product. They inherit common properties from their product family class.

For example, in Fig. 2, the Ash_decking_board product family class that is the superclass of the product variant class AD_20x95 includes property restrictions over hasFinishing object property and value restrictions on the hasUsage, hasThermalModification, and hasWoodSpecies data properties. In Fig. 2, we use the format of the ontology editor Protégé[6] to illustrate property restrictions as it is easy to read and short.

Ash_decking_board defines the class of products as the set of individuals that are linked to a finishing option by the hasFinishing property by using the cardinality restriction, which specifies that exactly one element can be in this relation. In addition, this class expression says that the class contains individuals that are connected by the hasFinishing property with an individual lacquering or oiling. In the similar way the specific class expressions are defined for the product variant class AD_20x95 for the hasProfile and the hasSuitableClip object properties.

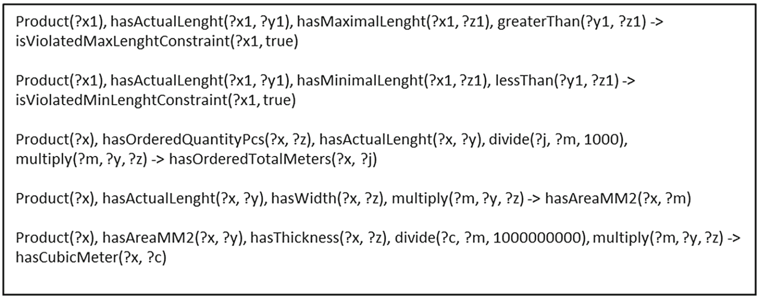
Using such principles of construction of class expressions makes it possible to use a DL reasoner to automatically infer predefined data property values for an ordered product as well as to check if an individual is expressing an ordered product consistent with ontology (i.e. does it satisfy the conditions given in the class expression).

4.4 Constraints as SWRL Rules and Reasoning

Class expressions are a convenient way to represent constraints in OWL. However, OWL [9] is not able to describe all relations needed to express constraints. The expressivity of OWL can be extended by adding SWRL [4] rules to ontology. SWRL rules are Horn clause like rules in what atoms can be basically of the form C(x) and P(x, y), where C is an OWL description, P is an OWL property, and x, y are either variables, OWL individuals or OWL data values [4]. SWRL includes a number of built-in predicates for individuals to manipulate with data values.

We define the following data properties to capture constraints for violation of maximum and minimum lengths given in the corresponding product class definition: isViolatedMaxLenghtConstraint and isViolatedMinLenghtConstraint. These data properties are used in rules to check whether the corresponding constraint was violated or not. The reasoner Pellet [11] asserts values for these data properties according to the SWRL rules that represent constraints for violation of maximum and minimum lengths. The corresponding rules are presented in Fig. 3.

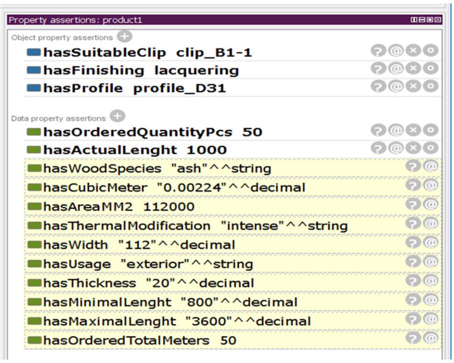
[6] https://protege.stanford.edu/.



```
Product(?x1), hasActualLenght(?x1, ?y1), hasMaximalLenght(?x1, ?z1), greaterThan(?y1, ?z1) ->
isViolatedMaxLenghtConstraint(?x1, true)

Product(?x1), hasActualLenght(?x1, ?y1), hasMinimalLenght(?x1, ?z1), lessThan(?y1, ?z1) ->
isViolatedMinLenghtConstraint(?x1, true)

Product(?x), hasOrderedQuantityPcs(?x, ?z), hasActualLenght(?x, ?y), divide(?j, ?m, 1000),
multiply(?m, ?y, ?z) -> hasOrderedTotalMeters(?x, ?j)

Product(?x), hasActualLenght(?x, ?y), hasWidth(?x, ?z), multiply(?m, ?y, ?z) -> hasAreaMM2(?x, ?m)

Product(?x), hasAreaMM2(?x, ?y), hasThickness(?x, ?z), divide(?c, ?m, 1000000000), multiply(?m, ?y, ?z) ->
hasCubicMeter(?x, ?c)
```

Fig. 3. SWRL rules (variables in rules are denoted using a question mark as a prefix)

We also developed some rules for lumber calculations as for the production and thermo-kiln management several measurements of lumber quantity are used. In general, measures of a board are given in millimeters. The reasoner fires rules and asserts data property values to the properties hasAreaMM2, hasCubicMeter, hasOrderedTotalMeters that are used for production orders and optimal packing of boards into thermo-kiln. In Fig. 4 we see the description of the individual named product1 that includes object and data property assertions related to the product order (marked with bold) and those that are asserted by the reasoner (marked with yellow background). The reasoner asserted data property values that correspond to standard parameters of the product and its product family as well as values that are asserted by rules.



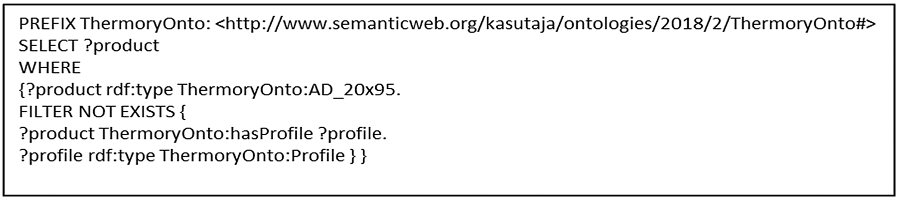
hasSuitableClip clip_B1-1
hasFinishing lacquering
hasProfile profile_D31

hasOrderedQuantityPcs 50
hasActualLenght 1000
hasWoodSpecies "ash"^^string
hasCubicMeter "0.00224"^^decimal
hasAreaMM2 112000
hasThermalModification "intense"^^string
hasWidth "112"^^decimal
hasUsage "exterior"^^string
hasThickness "20"^^decimal
hasMinimalLenght "800"^^decimal
hasMaximalLenght "3600"^^decimal
hasOrderedTotalMeters 50

Fig. 4. Reasoning results showing data property assertions (Color figure online)

4.5 Completeness of Descriptions

As mentioned in Sect. 4.2, to ensure that the description of a specific product variant in Abox is complete with regard to certain variant parameters we need to use Closed World Assumption (CWA), which assumes that the specification information is complete. However, CWA reasoning and its combinations with the OWA are not well supported by now. Therefore, we propose to use SPARQL queries [10] for checking completeness of descriptions of individuals as depicted in Fig. 5.



```
PREFIX ThermoryOnto: <http://www.semanticweb.org/kasutaja/ontologies/2018/2/ThermoryOnto#>
SELECT ?product
WHERE
{?product rdf:type ThermoryOnto:AD_20x95.
FILTER NOT EXISTS {
?product ThermoryOnto:hasProfile ?profile.
?profile rdf:type ThermoryOnto:Profile } }
```

Fig. 5. A SPARQL query example

This SPARQL query returns the list of individuals of the AD_20x95 class that do not have link via the hasProfile object property to any individual of the Profile class.

4.6 Lessons Learnt and Future Visions

Our experience of using the combination of OWL ontologies, SWRL and SPARQL to solve the product variant management problem described in this paper shows that combining CWA and OWA is not very convenient to work with in this framework. OWL is good for the description of the model and SWRL for the expression of additional constraints of the model. DL reasoning is well suited for validation of the model during the design time. However, using SPARQL queries (or query templates) for checking completeness of descriptions of Abox individuals (i.e. CWA) before using DL reasoning (i.e. OWA) in order to check correctness of descriptions of individuals wrt to the model is not convenient. Main reason is that OWA does not make it possible to check integrity constraints, such as whether a property has a value or object property has a link to an individual, etc. To overcome this limitation, for each affected object property we need to create a corresponding SPARQL query and run it to get the resulting set of individuals satisfying the given criteria (e.g. see Fig. 5).

In order to make our approach simpler and prepare it for an industrial use we are seeking for possibilities related to CWA that are offered by SPARQL inference notation SPIN [6] and its new development Shapes Constraint Language (SHACL) that is W3C recommendation since 2017[7]. SHACL allows expressing rules and checking integrity constraints that individuals need to satisfy as well as includes possibilities for expressing mathematical computations.

We are planning to combine both OWL ontology and SHACL statements in order to make our approach to meet industrial needs. This may lead us to the method, where we exclude SWRL rules and use only OWL ontologies and SHACL. SHACL can be integrated with SPARQL if necessary.

From the business point of view, we see several applications of this approach in the product variant management system in Thermory and in other enterprises. In addition, Thermory's B2B site can benefit from this ontology enabling to provide valid variant parameter value options and combinations for a customer to choose from. Using it within material resource planning is foreseen but this requires the extension of ontology with knowledge about material consumption.

[7] https://www.w3.org/TR/shacl/.

5 Conclusions

This paper presented the approach to expressing knowledge of variants of thermally modified wood products for solving product variant management problems in Thermory AS. We provided the ontology based representation of possible thermally modified product variants and used it for resolving inconsistencies in the ontology as well as in checking consistence between product variants and customer order requirements. We combined OWL and SWRL to represent constraints on valid product variants. We suggest using SPARQL to check the completeness of the description of an individual product variant before checking whether this individual is consistent with ontology.

The approach is general and enables to use its principles in many industries where product variant management is important issue. The benefits of the ontology based approach comparing to database based solution lie in the fact that class expressions and rules enable to express the same knowledge more efficiently and to reduce combinatorial complexity of describing and using of the product variant knowledge.

Acknowledgement. This work was partially supported by the Institutional Research Grant IUT33-13 of the Estonian Research Council.

References

1. Acher, M., Collet, P., Lahire, P., France, R.: Comparing approaches to implement feature model composition. In: Kühne, T., Selic, B., Gervais, M.-P., Terrier, F. (eds.) ECMFA 2010. LNCS, vol. 6138, pp. 3–19. Springer, Heidelberg (2010). https://doi.org/10.1007/978-3-642-13595-8_3
2. Genesereth, M.R., Fikes, R.E.: Knowledge Interchange Format Reference Manual. Technical report, Computer Science Department, Stanford University (1992)
3. Gruber, T.R.: Ontolingua: A Mechanism to Support Portable Ontologies. Technical report, Stanford University (1992)
4. Horrocks, I., et al.: SWRL: A Semantic Web Rule Language Combining OWL and RuleML. https://www.w3.org/Submission/SWRL. Accessed 05 Mar 2018
5. Ines, C., Crepinšek, M., Kosar, T., Mernik, M.: Ontology driven development of domain-specific languages. Comput. Sci. Inf. Syst. **8**(2), 317–342 (2011)
6. Knublauch, H., Hendler, J.A., Idehen, K.: SPIN - Overview and Motivation. https://www.w3.org/Submission/spin-overview/. Accessed 15 May 2018
7. Krueger, C., Clements, P.: Enterprise feature ontology for feature-based product line engineering and operations. In: Proceedings of SPLC 2017, 10 p. ACM, Spain (2017)
8. McGuinness, D., Wright, J.R.: An industrial - strength description logic-based configurator platform. IEEE Intell. Syst. **13**(4), 69–77 (1998)
9. Motik, B., et al.: OWL 2 Web Ontology Language: Structural Specification and Functional-Style Syntax. http://www.w3.org/TR/owl2-syntax. Accessed 04 Mar 2018
10. Patel-Schneider, P.F., McGuinness, D.L., Brachman, R.J., Resnick, L.A.: The CLASSIC knowledge representation system: guiding principles and implementation rationale. SIGART Bull. **2**(3), 108–113 (1991)
11. Sirin, E., Parsia, B., Grau, B.C., Kalyanpur, A., Katz, Y.: Pellet: A Practical OWL-DL Reasoner. Technical report, CS 4766, University of Maryland, College Park (2005)

12. Soininen, T., Tiihonen, J., Mannisto, T., Sulonen, R.: Towards a general ontology of configuration. Artif. Intell. Eng. Des. Anal. Manuf. AI EDAM **12**(4), 357–372 (1998)
13. SPARQL 1.1 W3C Recommendation Page. https://www.w3.org/TR/sparql11-overview/. Accessed 04 Mar 2018
14. Xuanyuan, S., Li, Y., Patil, L., Jiang, Z.: Configuration semantics representation: a rule-based ontology for product configuration. In: Proceedings of SAI Computing Conference, pp. 734–741. IEEE (2013)
15. Yang, D., Dong, M., Miao, R.: Development of a product configuration system with an ontology-based approach. Comput. Aided Des. **40**(8), 863–878 (2008)

The Knowledge Increase Estimation Framework for Integration of Ontology Instances' Relations

Adrianna Kozierkiewicz(✉) and Marcin Pietranik

Faculty of Computer Science and Management, Wroclaw University of Science
and Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
{adrianna.kozierkiewicz,marcin.pietranik}@pwr.edu.pl

Abstract. The previous authors' research showed that it is not only possible, but also profitable to estimate a potential growth of a level of knowledge that appears during an integration of ontologies. Such estimation can be done before the eventual integration procedure (or at least during such) which makes it even more valuable, because it allows to decide if a particular integration should be performed in the first place. Until now, authors of this paper prepared a formal framework that can be used to estimate the knowledge increase on the level of concepts, instances and relations between concepts. This paper is devoted to the level of relations between instances.

Keywords: Ontology · Integration · Knowledge increase

1 Introduction

With a growing number of available ontologies, their diversity and sizes, an efficient method of performing their integration is invaluable. However, no matter how efficient and well designed the integration algorithm, with large ontologies that are required to be merged, a method of estimating whether or not such integration will result in profitable outcomes can become useful.

We understand the integration itself as a task defined as follows: *for given n ontologies $O_1, O_2, ..., O_n$ one should determine an ontology O^* which is the best representation of given input ontologies.* Assuming that $\tilde{O}$ is a set containing all possible ontologies of interest this task can be achieved using a function σ with a signature $2^{\tilde{O}} \rightarrow \tilde{O}$ that accepts as an input any subset of $\tilde{O}$ and returns an outcome of the integration, denoted as O^*. Obviously, if only one ontology is given as an input, it is also returned as an output. Formally, $\sigma(O_1) = O_1$ for a selected $O_1 \in \tilde{O}$.

Our research are not focus on an issue of creating algorithms of the ontology integration (so a materialisation of the aforementioned function σ). However, in order to preserve the integrity of the paper, we have developed an integration algorithm that will be used to illustrate our ideas. As a foundation of the ontology integration algorithm, we tested and developed a procedure taken from the

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 172–186, 2018.
https://doi.org/10.1007/978-3-319-97571-9_15

literature [14]. Thus, we can concentrate on determining a formal framework that could be used to estimate a potential growth of knowledge that can be gained when ontologies are merged.

Going to the merits, by *"growth of knowledge"* we understand an indicator that shows whether or not the integration ends with a profitable outcome in the potential growth of knowledge point of view. For example, when two input structures contain the same knowledge, then nothing is gained from their integration. On contrary, if contents of input ontologies are entirely separate, then their merging can be invaluable.

Formally the task of creating a tool that allows such estimation can be defined as follows: *for given ontologies $O_1, O_2, ..., O_n$ from the set $\tilde{O}$ and an integration function σ one should determine a function Δ with a signature $\Delta : 2^{\tilde{O}} \to R$ that represents the increase of knowledge between input ontologies and a result of their merging $\sigma(O_1, O_2, ..., O_n) = O^*$*. Obviously, such approach can be decomposed into several subtasks that cover a wide array of ontology elements, namely: concepts, relations among them, their hierarchy, their instances and relations connecting instances.

In our previous publications [7–9], we developed a set of methods that allow to estimate the growth of knowledge during the integration of concepts (using a function Δ_C) and their hierarchy (using a function Δ_H), relations (using a function $\Delta_{R,C}$), and instances (using a function Δ_I). The missing piece is the estimation of knowledge that can be achieved through the integration of ontologies on a level of relations that described connections between concepts' instances. This level focuses on expressing how particular instances of the defined concepts interact with each other. In contrast to the level of concepts' relations (that we have covered in [9]) that is used to describe possible connections that may occur (e.g. *a man can be a husband of a woman*) the level of instances' relations focus of particular connections (e.g. *Joe is a husband of Jane*). Merging such statements may entail several difficulties concerning properties of relations such as their reflexivity, asymmetry or transitivity, that didn't appear when the integration of concepts relations has been considered.

For clarity, we assume that we deal only with the integration of only two ontologies. Therefore, the main contribution of the following paper can be formally defined as follows: For given two ontologies O_1 and O_2 that both contain sets representing relations between their instances denoted as R^{C_1} and R^{C_2} one should determine a function Δ_R with a signature $\Delta_R : \tilde{O} \times \tilde{O} \to [-1, \infty)$ representing an estimation of a potential growth of knowledge that can appear during the integration of O_1 and O_2 on the level of instances' relations.

The article is organised as follows. The next section contains a short description of a similar research found in the literature. In Sect. 3 the basic notions used throughout the paper are given. Section 4 formally describes a sought function Δ_R in terms of postulates that it must comply to. Eventually it ends with an algorithm that we have developed to calculate its values. It is then statistically analysed in Sect. 5. The paper ends with Sect. 6 which provides some conclusions and a brief overview of our upcoming work.

2 Related Works

Estimating the effectiveness of the ontology integration has not been widely investigated. In the literature it is possible to find some measures which allow to calculate how efficient the integration process is, however none of this research take into the consideration the potential growth of knowledge as the result of merging two or more ontologies.

Precision and recall with respect to a reference mapping are the most popular methods of measuring the quality of ontology matching [12,14] which can be formalised as a problem of an information distance metric like in [15] or in [6]. Some modifications of these measures have been proposed by [5]. Euzenat also provided a semantics for alignments based on the semantics of ontologies and has designed semantic precision and recall measures. The definition of these measures are independent from the semantics of ontologies. Such approach requires the use of logical reasoning, where both correspondences and ontologies are considered.

In [6] authors have demonstrated a novel measure named link weight that uses semantic characteristics of two entities and Google page count to calculate an information distance similarity between them. The proposed measure has been used to align ontologies semantically. In [4] authors have evaluated a wide range of string similarity metrics, along with string preprocessing strategies such as removing stop words and considering synonyms, on different types of ontologies. Additionally, the most efficient metrics for merging process have been pointed out. In [11] some ontology metrics used to measure distance between ontologies' semantics, rather than ontological structures are proposed. Those cohesion metrics have been: Number of Ontology Partitions (NOP), Number of Minimally Inconsistent Subsets (NMIS) and Average Value of Axiom Inconsistencies (AVAI).

In [13] authors introduced quality measures that are based on the notion of mapping incoherence that can be used without a reference mapping. This measure provides a strict upper bound for the precision of a mapping and can therefore be used as a guideline for estimating the performance of matching systems.

Ceusters [3] developed a metric, which is designed to allow assessment of the degree to which the integration of two ontologies yields improvements over either of the input ontologies. Authors noticed the fact, that input ontologies can contain some mistakes. However this paper contains only theoretical considerations about some factors which can be used to assess the adequateness of both the original ontologies and the results of matching or merging.

Some papers are devoted to measuring the quality of a single ontology. In [2], a tool called Ontology Auditor has been described. Authors have developed a suite of metrics that assess the syntactic, semantic, pragmatic, and social aspects of the concerned topic. Authors have designed many metrics like: overall quality, syntactic quality, lawfulness, richness, semantic quality, interpretability, consistency, clarity, pragmatic quality, comprehensiveness, accuracy, relevance, social quality, authority, history and described each of them.

OntoQa [17] is another model that analyses ontology schemas and their populations and describes them through a well defined set of metrics. Authors defined two categories of metrics. The schema metrics address the design of the ontology which indicate the richness, width, depth, and inheritance of an ontology schema. The instance metrics were grouped into two categories: knowledge base metrics and class metrics. The former describe the knowledge base as a whole. The latter which describe the way each class that is defined in the schema is being utilised in the knowledge base.

The ROMEO [19] methodology identifies requirement that an ontology must satisfy and maps the requirements to evaluation measures like consistency, conciseness, completeness, coverage, correctness, clarity, expandability, minimal ontological commitment. Similarity functions (ontology evaluation approaches) has been developed in other systems like: OntoClean [18] or OntoMetric [10] however, the mentioned solution evaluate the quality of the single or set of ontologies and does not consider the integration them.

All of the described metrics have many disadvantages. Many of them are calculated in separation with each other and none of the metric can give a big picture of the performed integration. Some measures are extracted from information retrieval field and do not consider the expressive structure of ontologies, while other assess only a single ontology and it cannot be applied in the estimation of the quality of the ontology integration process.

In this paper we propose measures that do not have flaws described above. It is devoted to the estimation of the potential increase of knowledge. The proposed method allows to decide about the profitability of the eventual integration process, which is a continuation of our previous research presented in [7–9]. Until now we have developed methods of the estimating the knowledge increase during the integration of ontologies on the level of concepts, instances and relations and hierarchies of concepts. In this article we focus on the integration of relations that occur between instances.

3 Basic Notions

We define a real world using a pair *(A, V)*, in which A is a set of attributes that can be used to describe objects taken from some topic and V denotes a set of valuations of these attributes. Formally, if V_a denotes a domain of an attribute a, a following condition is met: $V = \bigcup_{a \in A} V_a$. An ontology is a tuple:

$$O = (C, H, R^C, I, R^I) \tag{1}$$

where C is a set of concepts, H is concepts' hierarchy, R^C is a set of relations between concepts $R^C = \{r_1^C, r_2^C, ..., r_n^C\}$, $n \in N$, $r_i \subset C \times C$ for $i \in [1, n]$, I denotes a set of instances' identifiers and $R^I = \{r_1^I, r_2^I, ..., r_n^I\}$ symbolises a set of relations between concepts' instances. Every relation from the set R^C has a complementary relation from the set R^I. In other words, a relation $r_j^C \in R^C$ is a set containing descriptions of potential connections that may occur between instances of concepts from the set C, while $r_j^I \in R^I$ contains definitions of actually materialised connections. For example, the set R^C may contain relations *is_husband* or *is_wife* and in one can find R^I statements that *Dale is a husband of Laura or that Jane is a wife of David.* Obviously, $|R^C| = |R^I|$.

Concepts taken from the set C are defined as quadrupoles $c = (id^c, A^c, V^c, I^c)$, where id^c is an identifier of a concept c, A^c is a set of its attributes, V^c is a set attributes domains (formally: $V^c = \bigcup_{a \in A^c} V_a$) and I^c is a set of particular concepts' instances. We can write $a \in c$ which denotes the fact that the attribute a belongs to the concept's c set of attributes A^c. An ontology is called *(A, V)-based* if the conditions $\forall_{c \in C} A^c \subseteq A$ and $\forall_{c \in C} V^c \subseteq V$ are both met. As aforementioned in Sect. 1, a set of all (A, V)-based ontologies is denoted as $\tilde{O}$.

Given a concept c, we define its' instances as a tuple $i = (id^i, v^i_c)$. id^i is an identifier and v^i_c is a function with a signature: $v^i_c : A^c \to V^c$. Using a consensus theory [14], the function v^i_c can be interpreted as a tuple of type A^c.

A set of instances from the Eq. 1 is defined below:

$$I = \bigcup_{c \in C} \{id^i | (id^i, v^i_c) \in I^c\} \tag{2}$$

We write $i \in c$ to express that an instance with an identifier i belongs to a concept c.

In order to simplify operations on sets, we define an auxiliary notion of a set *Ins(c)* containing identifiers of instances assigned to concept c. Formally:

$$Ins(c) = \{id^i | (id^i, v^i_c) \in I^c\} \tag{3}$$

Complementary, Ins^{-1} denotes a helper function that designates concepts containing a given instance's identifier. It has a signature $Ins^{-1} : I \to 2^C$ and is defined as follows:

$$Ins^{-1}(i) = \{c | c \in C \wedge i \in c\} \tag{4}$$

Relations from the set R^C acquire semantics using L^R_s which is a sublanguage of the sentence calculus. This is accomplished using a function $S_R : R^C \to L^R_s$. Such approach allows to define criteria for relationships between relations:

- *equivalency* between relations r and *r'* (denoted as $r \equiv r'$) appears only if a sentence $S_R(r) \iff S_R(r')$ is a tautology
- a relation *r'* is more general than the relation r (denoted as $r' \leftarrow r$) if $S_R(r) \implies S_R(r')$ is a tautology
- *contradiction* between relations r and *r'* (denoted as $r \sim r'$) is true only if a sentence $\neg(S_R(r) \wedge S_R(r'))$ is a tautology.

As mentioned earlier, relations from the set R^C define what concepts instances can be connected with each other, while R^I defines what actually is connected. To denote this requirement, we use the same index of relations taken from both sets - a relation $r^I_j \in R_I$ contains instance pairs that are mutually connected by a relation $r^C_j \in R^C$. Below, we define a set of formal criteria that these sets must meet:

1. $r_j^I \subseteq \bigcup_{(c_1,c_2)\in r_j^C} (Ins(c_1) \times Ind(c_2))$
2. $(i_1, i_2) \in r_j^I \implies \exists(c_1, c_2) \in r_j^C : (c_1 \in Ins^{-1}(i_1)) \wedge (c_2 \in Ins^{-1}(i_2))$ which describes that two instances may be connected by some relation only if there is a relation connecting concepts they belong to
3. $(i_1, i_2) \in r_j^I \implies \neg\exists r_k^I \in R^I : ((i_1, i_2) \in r_k^I) \wedge (r_j^C \sim r_k^C)$ which concerns a situation in which two instances cannot be connected by two contradicting relations (e.g. John cannot be simultaneously a husband and a brother of Jane)
4. $(i_1, i_2) \in r_j^I \wedge \exists r_k^I \in R^I : r_k^C \leftarrow r_j^C \implies (i_1, i_2) \in r_k^I$ which denotes that if two instances are in a relation and there exists a more general relation, then they are also connected by it (e.g. if John is a father of David, then he is obviously also his parent).

Relations connecting concepts are used to define what types of object can interact with each other. On this level it is not necessary to define specific properties of relations using their semantics originating from L_s^R. On the level of instances relations, that actually materialise relations the actual properties of relations come into play.

For example, defining that *a man* can *be a brother* or *be a husband* of *a woman* is quite simple on the level of concepts. On the level of instances it is crucial to also express that *John* cannot simultaneously be a husband and a brother of *Jane*. Merging two sets (that are used to define relations according to the Eq. 1) which contain pairwise excluding knowledge would lead to inner inconsistencies within an ontology that is a result of the integration. Therefore, in the L_s^R we distinguish two elements *is_asymmetric* and *is_transitive*, that for some selected relation r can be used to describe its following properties:

- $(S_R(r) \implies is\_asymmetric) \iff \forall\,(a, b) \in r : \neg\exists\,(b, a) \in r$
- $(S_R(r) \implies is\_transitive) \iff \forall\,(a, b, c) \in C : (a, b) \wedge (b, c)\,\exists\,(a, c) \in r$

To simplify the notation, in subsequent parts of the paper, we will use predicates $is\_asymmetric(r)$ and $is\_transitive(r)$.

In our considerations we do not include relations that are symmetric. The reason why is a property of such relation - if two symmetric relations (that are in fact sets) are summed, the resulting set will also be symmetric. As a result, no conflicting statements about connected instances will emerge. The same situation occurs when integrating relations that are reflexive or irreflexive. Therefore, we also do not include them in our framework.

4 Integration of Ontology Instances' Relations

In this section, an algorithm for the ontology integration on instances' relations level will be presented. As an input, it requires to ontologies that are defined according to Eq. 1. As a result, it returns sets of integrated relations (for both levels of concepts and instance) and a final estimation of the knowledge increase on the level of instance gained during the conducted integration. As stated in Sect. 1, this estimation is a value of a function $\varDelta_R$ that for two ontologies $O_1 = (C_1, H_1, R^{C_1}, I_1, R^{C_1})$ and $O_2 = (C_2, H_2, R^{C_2}, I_2, R^{C_2})$ has a signature $\varDelta_R : R^{C_1} \times R^{C_2} \rightarrow [-1, \infty]$.

Assuming that both ontologies contain only one relation (formally: $R^{C_1} = \{r_1^C\}, R^{I_1} = \{r_1^I\}, R^{C_2} = \{r_2^C\}, R^{I_2} = \{r_2^I\}$), the function $\varDelta_R$ is described by the following postulates:

1. $\varDelta_R = -1 \iff (r_1^C \equiv r_2^C) \wedge is\_asymmetric(r_1^C) \wedge \forall_{(a,b) \in r_1^I} \exists (b, a) \in r_2^I$
2. $\varDelta_R = 0 \iff (r_1^C \equiv r_2^C) \wedge (r_1^I \cup r_2^I = r_1^I \cap r_2^I)$
3. $\varDelta_R = 1 \iff \neg((r_1^C \equiv r_2^C) \vee (r_1^C \leftarrow r_2^C) \vee (r_1^C \sim r_2^C))$
4. $\varDelta_R \in [1, \infty] \iff (r_1^C \equiv r_2^C) \wedge is\_transitive(r_1^C)$

The first postulate concerns an issue in which two relations are equivalent, but asymmetric. In such situation the resulting relation must also be asymmetric, but also cannot contain pairs of instances that interfere with the asymmetry. If all of the instances' pairs from two relations do so, then the $\varDelta_R$ must express that the integration not only do not increase the knowledge, but actually causes its loss.

The second postulate illustrate the repetitive knowledge in two ontologies. In such situation $\varDelta_R$ should be equal to 0, which expresses the situation where nothing is gained from the conducted integration.

The third postulate defines a situation in which two relations expresses two completely different interactions that may occur between instances. These interactions do not interfere with each other (which may result in knowledge loss), but do not entail the emergence of any new knowledge about instances' relations. This kind of synergy is expressed using the fourth postulate.

The eventual shape of the proposed method is presented on Algorithm 1. At first, in lines 1 to 3, the algorithm creates two empty sets for the results and initialises the knowledge increase estimator $\varDelta_R$. The backbone of the algorithm (line 4) is an iteration through a Cartesian product of two sets of concepts relations.

In each loop, the algorithm at first (in line 5) checks if the two relations that are currently analysed are equivalent. If this is the case, then the algorithm attaches to the final result a sum of processed relations (a single relation that contains elements of both inputs) and adds to the knowledge increase estimator $\varDelta_R$ an ordinary Jaccard's similarity between the two in order to express how the conducted integration enriches the overall knowledge. This is done in lines 6 to 10.

However, the equivalency requires to check if the resulting relation (created in lines 6 and 7) is asymmetric or transitive. The former property may entail potential loss of knowledge due to the fact that in the resulting ontology two instances cannot be connected symmetrically. If in the two input ontologies two instances are connected by the equivalent relations, but in the different order, this can cause that the knowledge coming from the fact that the two instances are connected becomes uncertain. To avoid such situation, the algorithm removes (in lines 16 and 17) both connections and decreases the value of the estimator δ_R.

In the next step (line 23), the algorithm checks the transitivity of the resulting relation created in lines 6 and 7. This situation may entail the emergence of a new knowledge. For example, for the relation *is family* a situation in which one of the input ontologies contains a pair *(John, Steven)* and the second contains a pair *(Steven, Wilson)* should result that after the integration its output should also include a pair *(John, Wilson)*. This knowledge has not been present in the input ontologies, but emerges thanks to the conducted integration. In other words, the integration of ontologies is a synergy, where its output is something more than a strict sum of the input. Therefore, the algorithm increase the knowledge estimator δ_R that can acquire values larger than 1, meaning that it is not a metric, but it may indicate that a new knowledge has been created.

The subsequent part of the algorithm (lines 31–36) copes with a situation in which one of the processed relations is more general than the other. In such situation, the algorithm adds to the resulting ontology both relations, but also expands the less specific relation with the elements of the second relation. This indicates that some of the knowledge is gained thanks to the integration, but some is actually repeated in both input ontologies. Therefore, the knowledge increase estimator δ_R is increased only by the relations that are not present in both relations.

The next stage of the algorithm handles a situation when two relations are contradicting. Both relations are included in the final ontology, but this issue may result in knowledge decrease, due to the fact that two instances cannot be simultaneously connected by such relations. For example, *John* cannot be both *a brother* and *a husband* of *Jane*. If such pairs are found, then they are removed from the resulting ontology (lines 40 and 41) and the knowledge estimator Δ_R is decreased accordingly (line 42).

The last part of the algorithm (lines 44 to 49) covers the simple case when two relations express completely different knowledge concerning how instances interact with each other. In this situation, both relations are added to the final ontology and the estimator Δ_R is increased with a maximal value (equal to 1), because both input ontologies bring new knowledge to the final result.

Eventually, the algorithm returns created sets of relations and a mean knowledge increase (line 52) that has been acquired due to the conducted integration.

5 Evaluation of the Proposed Formula

Our research focuses on creating a new methodology of estimating a potential growth of knowledge during the ontology integration process. Therefore, there is no benchmark dataset that could be used to prove the correctness of our ideas. To verify our ideas, we used a statistical analysis of data obtained from questionnaire that contained 20 questions showing results of the integration of two ontologies on the instances' relation level[1]. The human judgment is the popular and approved methodology [16] for this kind of verification.

[1] https://goo.gl/iktxsK.

Algorithm 1. Ontology integration and knowledge increase estimation on relation level

```
Require: $O_1 = (C_1, H_1, R^{C_1}, I_1, R^{C_1}), O_2 = (C_2, H_2, R^{C_2}, I_2, R^{C_2})$;
$R^{C^*} = \phi$;
$R^{I^*} = \phi$;
$\Delta_R = 0$;
for $(r_i^{C_1}, r_j^{C_2}) \in R^{C_1} \times R^{C_2}$ do
    if $r_i^{C_1} \equiv r_j^{C_2}$ and $(r_i^{I_1} \cup r_j^{I_2}) \notin r^{I^*}$ then
        $r^{C^*} = r_i^{C_1} \cup r_j^{C_2}$;
        $r^{I^*} = r_i^{I_1} \cup r_j^{I_2}$;
        $R^{C^*} = R^{C^*} \cup \{r^{C^*}\}$;
        $R^{I^*} = R^{I^*} \cup \{r^{I^*}\}$;
        $\Delta_R = \Delta_R + 1 - \frac{|r_i^{I_1} \cap r_j^{I_2}|}{|r_i^{I_1} \cup r_j^{I_2}|}$;
        if $is\_assymetric(r^{C^*})$ then
            $count = 0$
            $s = |r^{I^*}|$
            for $(a, b) \in r^{I^*}$ do
                if $(b, a) \in r^{I^*}$ then
                    $r^{I^*} = r^{I^*} \setminus \{(b, a)\}$
                    $r^{I^*} = r^{I^*} \setminus \{(a, b)\}$
                    $count+ = 2$
                end if
            end for
            $\Delta_R = \Delta_R - 2 \cdot \frac{count}{s}$
        end if
        if $is\_transitive(r^{C^*})$ then
            for $(a, b) \in r^{I^*}$ do
                if $\exists c \in C^* : (b, c) \in r^{I^*}$ then
                    $r^{I^*} = r^{I^*} \cup \{(a, c)\}$
                end if
            end for
            $\Delta_R = \Delta_R + \frac{|r^* \setminus (r_i^{I_1} \cup r_j^{I_2})|}{|r_i^{I_1} \cup r_j^{I_2}|}$
        end if
    else if $r_i^{C_1} \leftarrow r_j^{C_2}$ then
        $r^{C^*} = r^{C^*} \cup \{r_j^{C_2}\}$
        $r^{C^*} = r^{C^*} \cup \{r_i^{C_1} \cup r_j^{C_2}\}$
        $r^{I^*} = r^{I^*} \cup \{r_j^{I_2}\}$
        $r^{I^*} = r^{I^*} \cup \{r_i^{I_1} \cup r_j^{I_2}\}$
        $\Delta_R = \Delta_R + \frac{|r_i^{I_1} \cap r_j^{I_2}|}{|r_i^{I_1}|}$
    else if $r_i^{C_1} \sim r_j^{C_2}$ and $r_i^{I_1} \notin r^{I^*}$ then
        $r^{C^*} = r^{C^*} \cup \{r_i^{C_1}\}$
        $r^{C^*} = r^{C^*} \cup \{r_j^{C_2}\}$
        $r^{I^*} = r^{I^*} \cup \{r_i^{I_1} \setminus r_j^{I_2}\}$
        $r^{I^*} = r^{I^*} \cup \{r_j^{I_2} \setminus r_i^{I_1}\}$
        $\Delta_R = \Delta_R + \frac{|r_i^{I_1} \setminus r_j^{I_2}|}{|r_i^{I_1}|} + \frac{|r_j^{I_2} \setminus r_i^{I_1}|}{|r_j^{I_2}|} - \frac{|r_i^{I_1} \cap r_j^{I_2}|}{|r_i^{I_1} \cup r_j^{I_2}|}$
    else
        if $r_i^{I_1} \notin r^{I^*}$ then:
            $r^{C^*} = r^{C^*} \cup \{r_i^{C_1}\}$;
            $r^{C^*} = r^{C^*} \cup \{r_j^{C_2}\}$;
            $r^{I^*} = r^{I^*} \cup \{r_i^{I_1}\}$;
            $r^{I^*} = r^{I^*} \cup \{r_j^{I_2}\}$;
            $\Delta_R = \Delta_R + 1$;
        end if
    end if
end forreturn $R^{C^*}, R^{I^*}, \frac{\Delta_R}{|R^{I^*}|}$;
```

The main aim of our experiments is demonstrated how the developed measure reflect a way people evaluate the knowledge increase. Prepared examples have covered all possible cases like: the integration of two equivalent relations, two contradicting relations, two asymmetric relations, two transitive relations, and situation where one relation is more general than other one. The instances' relations cannot be considered without a semantic meaning, therefore we didn't use any symbolic data. Additionally, we cannot use the real ontologies because the provided datasets are too big and it is very hard to process them manually by an expert. Some examples of prepared ontologies are presented in Figs. 1 and 2.

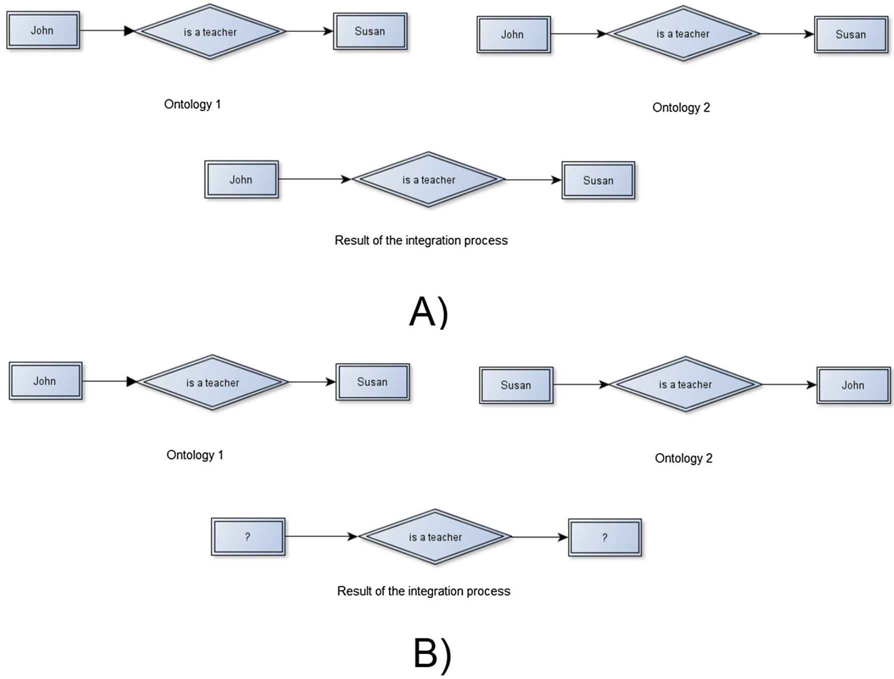


Fig. 1. Examples of knowledge increase during ontologies integration at instances' relations level; (A) $\Delta_R = 0$; (B) $\Delta_R = -100\%$.

The experiment has been divided in two parts. In the first one, we wanted to check the general trend. We have asked our responders about an overall opinion about knowledge change during the integration process. The responders could choose one from the three options: *the knowledge has grown, the knowledge remained the same, the knowledge has been lost (in other words - some semantic conflict occurred)*. In the second part the responders were asked to rate the level of knowledge change with the use of the scale range from -100% to ∞. This range corresponds with range $[-1, \infty]$, however the percentage values are more intuitive for human.

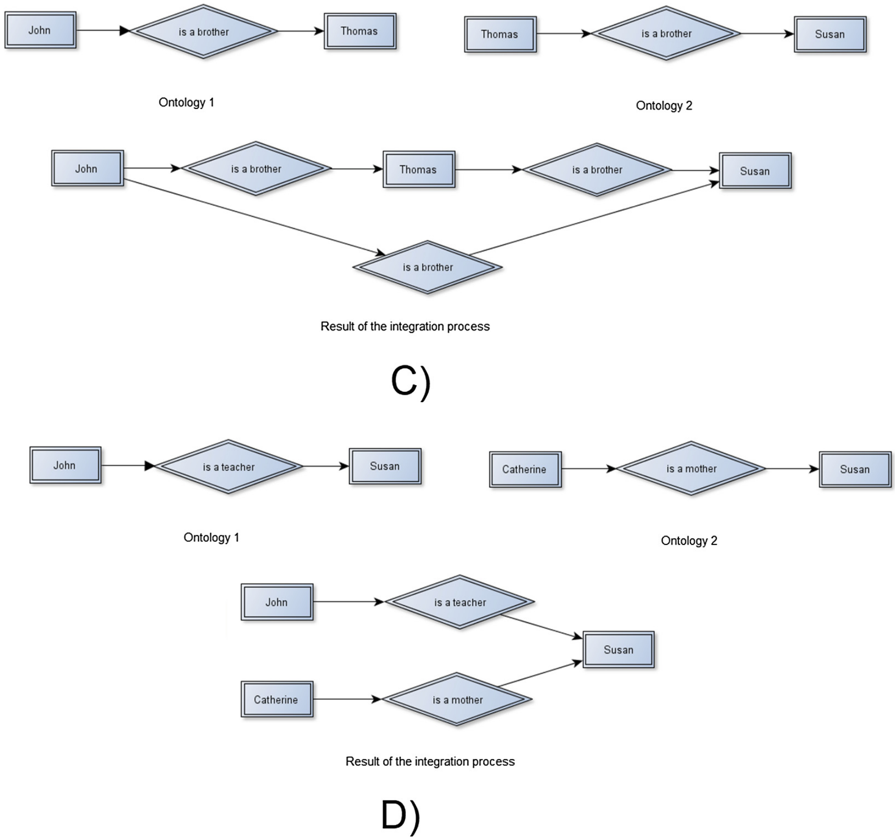


Fig. 2. Examples of knowledge increase during ontologies integration at instances' relations level; (C) $\Delta_R = 150\%$; (D) $\Delta_R = 100\%$.

Participants were not randomly selected - the survey was sent to a self-selected, biased population of people with technical science background. We decided to use that type of sampling because our responders needed to be familiar with issues related to databases, ontologies, instances and relations. The questionnaire was filled by 45 responders differing in age, sex, and educational status.

The obtained data have been pre-processed before a statistical analysis. The answers of our responders have been treated as experts' opinions and a median of their answers for each question has been calculated as the final estimation of the general trend and a value of knowledge increase estimation. In other words - based on the collected data, we have determined a consensus [14], as the final, common opinion. According to the literature it is possible to determine such consensus satisfying a 1- or 2-*optimality* criterion. For our purpose, we have chosen the 1-*optimality* postulate, which requires the result of the integration to be as near as possible to element of the input. The final summary of our analysis can

Table 1. The results of statistical analysis.

The type of experiment	The result	p-value
Cohen's Kappa coefficient-the general trend verification	$\kappa = 70.15\%$	0.000006
Correlation Coefficient test-the absolute agreement	0.579	0.0045
Correlation Coefficient test-the consistency	0.714	0.0045

be found in the Table 1. In the next subsection, some discussion about obtained results are presented.

5.1 The General Trend Verification

We had two samples to analyse. The first sample contained 20 elements, each determined as the consensus of expert's opinion referring to the general trend of the knowledge increase for the cases presented in the questionnaire. The second sample coming from the Algorithm 1.

All of the values are on nominal scale, therefore, the significance test for Cohen's Kappa coefficient has been used for the analysis. It was made with a significance level $\alpha = 0.05$. The agreement with a chance adjustment $\kappa = 70,15\%$ is smaller than the one which is not adjusted for the chances of an agreement and the *p-value* is equal 0.000006. Such result proves a statistical agreement between these two samples on the assumed significance level. It means that people can notice a general trend referring to the knowledge increase in the given integration task.

5.2 The Value of Formula Verification

As in the previous section, we had two samples to analyse. The first calculated as the consensus of experts' opinions and the second was created automatically using the formula presented in Algorithm 1. However, in this part of our experiment, our samples contained a value of potential growth of knowledge which occurs during the ontology integration process. For this purpose, we have checked the normality distribution of both samples using the Shapiro-Wilk test. For both samples *p-value* was greater than the accepted significance level $\alpha = 0.05$, therefore, we couldn't reject the null hypothesis, which states that both samples come from a normal distribution.

For the further analysis, we selected the Intraclass Correlation Coefficient test (ICC) [1]. It measures the strength of an inter-judge reliability – the degree of their's assessments concordance. We had two samples for which we have tested the consistency and the absolute agreement. For both test *p-value* was around 0.0045 which allows us to claim that values obtained from the Algorithm 1 and those based on experts' opinions are statistically concordant in the analysed population. However, the absolute agreement is not very high and is 0.579. The consistency has been calculated as 0.714. The gathered results are presented in Fig. 3. It could be seems that the respondent's answers do not achieve extreme points on the scale. However, each point on the left side of the Fig. 3 is consensus of 45 responders answers calculated in simply way as a median.

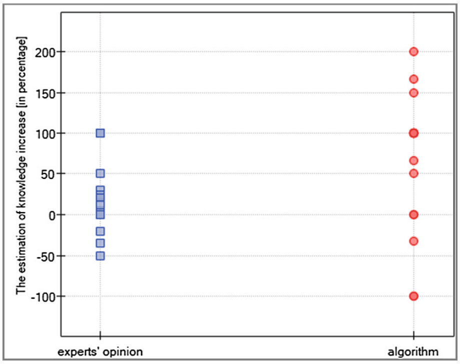


Fig. 3. The results of the experiment.

We can claim that the proposed method corresponds with a natural way people estimate the increase of knowledge. The integration process of instances' relations is not very intuitive, which makes it an interesting direction of upcoming research.

6 Future Works and Summary

The paper addresses the problem of the ontology integration on instances' relations level. Authors proposed the algorithm which allows to estimate a potential growth of knowledge during the merging process of two ontologies on instances' relations level.

The proposed method can be verified using several types of statistical tools used on the collected data, i.e. surveys, experimental studies and observations, among which a survey was selected for the stated purpose. The questionnaire containing 20 different scenarios has been prepared and presented to 45 volunteers. Based on the collected data, the consensus of experts' answers has been determined for each question. These results were compared with answers obtained by the execution of the proposed algorithm.

Statistical analysis (the Cohen Kappa method and ICC measure) allowed us to draw a conclusion that the created method of estimating a potential growth of knowledge is intuitive in a human experts way. The first experiment showed substantial agreement around 70%. It means that people can notice a general trend referring to the knowledge increase. The idea of relations of instances and their integration is not intuitive and it is hard to explain to people that are not familiar with the topic. However, the absolute agreement was nearly 58% in case of the comparison of the value of potential growth of knowledge. This agreement can be interpreted as fair.

In this paper, the method which estimates the objective growth of knowledge has been proposed. It means that our algorithm do not judge the disperse in the amounts of knowledge available in the input ontologies. In our upcoming publications, we would like to focus on subjective measures (from the point of view of the integrated ontologies) and conduct more experiments using real ontologies, which could bring more expressive conclusions.

Acknowledgments. This research project was supported by grant No. 2017/01/X/ST6/00491 from the National Science Centre, Poland

References

1. Bartko, J.J.: The intraclass correlation coefficient as a measure of reliability. Psychol. Rep. **19**(1), 3–11 (1966). https://doi.org/10.2466/pr0.1966.19.1.3
2. Burton-Jones, A., et al.: A semiotic metrics suite for assessing the quality of ontologies. Data Knowl. Eng. **55**(1), 84–102 (2005)
3. Ceusters W., Smith B.: Towards a realism-based metric for quality assurance in ontology matching. In: Proceedings of the 2006 Conference on Formal Ontology in Information Systems: Proceedings of the Fourth International Conference (FOIS 2006), pp. 321–332. IOS Press (2006)
4. Cheatham, M., Hitzler, P.: String similarity metrics for ontology alignment. In: Alani, H., et al. (eds.) ISWC 2013. LNCS, vol. 8219, pp. 294–309. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41338-4_19
5. Euzenat, J.: Semantic precision and recall for ontology alignment evaluation. IJCAI **7**, 348–353 (2007)
6. Jiang, Y., Wang, X., Zheng, H.T.: A semantic similarity measure based on information distance for ontology alignment. Inf. Sci. **278**, 76–87 (2014)
7. Kozierkiewicz-Hetmańska, A., Pietranik, M.: The knowledge increase estimation framework for ontology integration on the concept level. J. Intell. Fuzzy Syst. **32**(2), 1161–1172 (2017). https://doi.org/10.3233/JIFS-169116
8. Kozierkiewicz-Hetmańska, A., Pietranik, M., Hnatkowska, B.: The knowledge increase estimation framework for ontology integration on the instance level. In: Nguyen, N.T., Tojo, S., Nguyen, L.M., Trawiński, B. (eds.) ACIIDS 2017. LNCS (LNAI), vol. 10191, pp. 3–12. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-54472-4_1
9. Kozierkiewicz-Hetmańska, A., Pietranik, M.: The knowledge increase estimation framework for ontology integration on the relation level. In: Nguyen, N.T., Papadopoulos, G.A., Jędrzejowicz, P., Trawiński, B., Vossen, G. (eds.) ICCCI 2017. LNCS (LNAI), vol. 10448, pp. 44–53. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-67074-4_5
10. Lozano-Tello, A., Gomez-Perez, A.: OntoMetric: a method to choose the appropriate ontology. J. Database Manag. **15**(2), 1–18 (2004)
11. Ma, Y., Jin, B., Feng, Y.: Semantic oriented ontology cohesion metrics for ontology-based systems. J. Syst. Softw. **83**(1), 143–152 (2010)
12. Maleszka, M., Nguyen, N.T.: A method for complex hierarchical data integration. Cybern. Syst. **42**(5), 358–378 (2011)
13. Meilicke, Ch., Stuckenschmidt, H.: Incoherence as a basis for measuring the quality of ontology mappings. In: Proceedings of the 3rd International Conference on Ontology Matching, vol. 431. CEUR-WS. org (2008)

14. Nguyen, N.T.: Advanced Methods for Inconsistent Knowledge Management. Springer, London (2008). https://doi.org/10.1007/978-1-84628-889-0
15. Pietranik, M., Nguyen, N.T.: A Multi-atrribute based framework for ontology aligning. Neurocomputing **146**, 276–290 (2014). https://doi.org/10.1016/j.neucom.2014.03.067
16. Resnik, P.: Using information content to evaluate semantic similarity in a taxonomy. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, vol. 1, pp. 448–453 (1995)
17. Tartir, S., et al.: OntoQA: metric-based ontology quality analysis. http://lsdis.cs.uga.edu/library/download/OntoQA.pdf (2005). Accessed 22 Oct 2017
18. Welty, C., Guarino, N.: Supporting ontological analysis of taxonomic relationships. Data Knowl. Eng. **39**(1), 51–74 (2001)
19. Yu, J., Thom, J.A., Tam, A.: Requirements-oriented methodology for evaluating ontologies. Inf. Syst. **34**(8), 766–791 (2009)

Advanced Database Systems

Proposal of an Unrestricted Character Encoding for Japanese

Antoine Bossard[1](✉) and Keiichi Kaneko[2]

[1] Graduate School of Science, Kanagawa University,
2946 Tsuchiya, Hiratsuka, Kanagawa 259-1293, Japan
abossard@kanagawa-u.ac.jp

[2] Graduate School of Engineering, Tokyo University of Agriculture and Technology,
2-24-16 Nakacho, Koganei, Tokyo 184-8588, Japan

Abstract. The vast majority of characters used in Japanese are Chinese characters, which involve tens of thousands different glyphs. Due to this huge number of glyphs, database creation for character representation on computer systems has been an ongoing issue for years, and it has been actually since the early days of digital computing. Several character encodings have been described to allow the representation of character information, some specifically targeting Chinese characters, such as Big-5 and Shift-JIS, and others remaining general, such as Unicode. Yet, no matter the approach followed, it is still impossible to manipulate a large part of these characters as they are simply not covered by the current encoding solutions. Chinese characters feature various properties and relations, thus making it possible to classify them in a database according to several attributes. In this paper, we formally describe for such a large character database a structure in the form of a character encoding, thus aiming at addressing the concrete issue of character computer representation. It shall be shown that the proposed structure addresses the restrictions, such as coherency and glyph number, suffered by existing works. Finally, a database corresponding to the presented character encoding is practically assembled and visualised, demonstrating the advanced code structure.

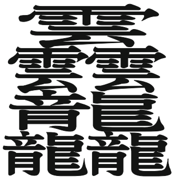
Keywords: Code · Information representation · Database · Glyph
Logogram · Symbol · Chinese

1 Introduction

For decades, various encodings have been proposed and used on computer systems. Such encodings can be considered as databases of thousands of entries, each entry identifying one character (glyph). While it is rather straightforward to design a character encoding to express the characters of the Latin alphabet, it is a very challenging issue in the case of Chinese characters. This can be explained by at least two facts: the number of characters involved is huge – and

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 189–201, 2018.
https://doi.org/10.1007/978-3-319-97571-9_16

unknown – and a lack of homogeneity regarding glyphs, a character possibly written differently depending on cultures and dialects.

The current situation is indeed appalling: as of today, it is impossible to input, let alone represent, numerous Chinese characters as they are not covered by the encodings implemented in computer systems. This is the case for example of the *otodo* character  famous for its high number of strokes. The aim of this research is to address this issue of information representation with respect to Chinese characters.

In concrete terms, the objective of this research is to propose an unrestricted character encoding for Chinese characters as used in Japanese. In this paper, we describe such a code that features the following characteristics:

- The code allows the collection of an unlimited number of glyphs.
- The code remains highly flexible: characters can be added (and removed if necessary) without any disturbance in the code.
- The code is easy to use: a character can be located with minimum effort.
- The code obviously forbids character duplicates.

Regarding the state of the art, it is important to mention that several character encodings are supported by modern computer systems, these encodings implementing two different approaches: the unified approach and the non-unified approach. The unified approach can be summarised as follows: all glyphs are represented by one encoding; this is the approach followed by Unicode [1]. In the particular case of Chinese characters, this means that the glyphs as found in Chinese, Japanese, Korean and so on are all gathered in one same place, and rendered similarly no matter the writing system (language); glyph stylistic differences are enforced by fonts. Even though this unified approach benefits from obvious advantages such as facilitated multi-language document manipulation, it has been the subject of criticism for many years (refer for instance to [2]). Unification itself is one of the raised issues: is it sound to unify characters of different cultures and writing systems, even though they have some parts in common? Another critical issue is the accessibility of the code: it is arguably tedious work to search, and if in luck to locate, one glyph as the unified writing systems do feature differences regarding character classification and ordering, these two issues being for example directly related to character pronunciation, which obviously differs from one language to another.

Non-unifying encodings include for instance Big-5 for Chinese [3], JIS for Japanese [4] and EUC-KR for Korean [5]. Even though not facing the aforementioned Unicode issues such as accessibility, these encodings are severely limited in that they cover a small portion of the Chinese characters (e.g., a few thousands, smaller than or hardly equal to 10,000 glyphs), prioritising the most frequent ones, thus leaving infrequent and rare ones impossible to be represented and thus becoming on the verge of extinction as our modern societies are relying always more on computer systems. In comparison, the IPA *mojikiban* database [6] lists almost 70,000 characters, which shows the gap between current encodings and actual needs. Of course, it can be argued that most of these uncovered characters are unused, or almost unused, but this should not become a reason for complete dismissal of these characters which do appear in various, possible ancient, texts. Authors ought to be able to use these as well, which is at the moment not possible, Unicode or not.

Finally, not directly related to character encoding but surely related to our proposal, the *Shikaku gōma* (四角號碼) Chinese character lookup method [7] relies on four digits (with a variant using five digits) to quickly find a character inside a list sorted according to these four digits. Notwithstanding the merits of this method, especially considering that it was introduced before digital computers, this classification does not rely nor retain character properties and thus lacks structuring. Also, prohibitively, it does not guarantee uniqueness of character codes, which thus makes it unsuitable as character encoding.

The rest of this article is organised as follows. First, several important properties of Chinese characters are recalled in Sect. 2. Then, the proposed encoding is detailed in Sect. 3, describing the code structure and character lookup. A database corresponding to the presented character encoding is practically assembled and visualised in Sect. 4, demonstrating the advanced code structure. Finally, this paper is concluded in Sect. 5.

2 Preliminaries

First, regarding terminology, the proposed encoding targets the Chinese characters as used in Japanese, also informally known as *kanji* characters. Yet, the described approach could also be applied to Chinese characters in general with only minor adjustments, since the characters in use are essentially the same. For the sake of conciseness, we shall in this paper simply refer to the code glyphs as (Chinese) characters. We define $\mathbb{J}$ the set of the Chinese characters as used in Japanese (more details on this can be found in our previous work; refer for instance to [8]).

Next, the properties of Chinese characters that are used hereinafter are recalled. Giving a formal character ontological description as in [8,9] is not our objective here; as just stated, we focus on the character properties that support defining the proposed encoding.

- First, a usual method to classify Chinese characters is to rely on their radicals. Each character has one unique radical, and there exist 214 radicals in the modern classification system.
- Second, each character is made of one or several strokes. Strokes are drawn in a precise order.
- Third, a character may have variants. In other words, there may exist several different ways to write a character.

3 Information Representation Methodology

The proposed encoding is detailed in this section, beginning with the description of the code structure and continuing with the presentation of formal tools to practically use the code.

3.1 Code Structure

Each character is mapped to a unique coordinate in a three-dimensional space. Concretely, the code $\mathcal{C}$ is organised according to a three-dimensional structure as follows.

X axis character radicals;
Y axis number of strokes (not counting the radical ones);
Z axis character variants (if any).

As we consider the standard 214 character radicals, the X axis spans the integers from 0 to 213, each radical being assigned a unique index i with $0 \leq i \leq 213$. Radical index assignment is performed conventionally, ordering radicals according to their stroke numbers in ascending order. Formally, let $\hat{\mathbb{J}}$ be the set of the 214 Chinese character radicals r_i $(0 \leq i \leq 213)$ as defined in Japanese. We define the radical indexing function $r : \hat{\mathbb{J}} \to \mathbb{N}$ which associates to a radical a unique positive integer (index). For that, we consider the ordered set $\bar{R}$ whose elements are those of $\hat{\mathbb{J}}$, and with the total order relation $r_i < r_j$ for any two distinct radicals $r_i, r_j \in \hat{\mathbb{J}}$ holding if and only if the radical r_i is listed before the radical r_j in the conventional radical ordering of Japanese dictionaries – the sequencing details are abbreviated here, refer for instance to the Kadokawa Shinjigen dictionary [10]. Therefore, assuming $\bar{R} = \left\{r_{p(0)}, r_{p(1)}, \ldots, r_{p(213)}\right\}$ in this order for some permutation p of the integers $0, 1, \ldots, 213$, the radical r_i $(0 \leq i \leq 213)$ is indexed to $r(r_i) = p(i)$.

Because of technical details, the Y axis is discussed later in this section; for now, it simply represents the number of strokes of characters, excluding the radical strokes. For example, the total number of strokes of the character 沖 is 7: precisely, 3 strokes for the radical (氵), and 4 other strokes (中); the Y coordinate is thus induced here by 4. It should be noted here that the radical 氵 is actually a radical variant of 水 of 4 strokes, which does not impact the way strokes are counted.

Regarding the character variants (Z axis), the character of coordinate 0 on the Z axis is called the "regular" (i.e., standard) variant. Because we are focusing on the Japanese writing system, it is natural to define the regular variant of a character as the one listed in the Ministry of Education's "Table of regular-use Chinese characters" [11], and in general the new form of a character if a new–old distinction is made. In short, we abide by the rules followed by Japanese modern dictionaries. It should be noted that the radical of a variant is naturally the same as that of the corresponding regular character. Finally, the variants of a regular character are not sorted, that is, for a regular character of coordinate $(x, y, 0)$, the characters of coordinates (x, y, z) with $z > 0$ are arbitrarily sequenced (i.e., unordered).

An example involving the characters 杉, 枝 both of radical 木 and of stroke numbers (not counting the radical strokes) 3 and 4, respectively, is illustrated in Fig. 1.

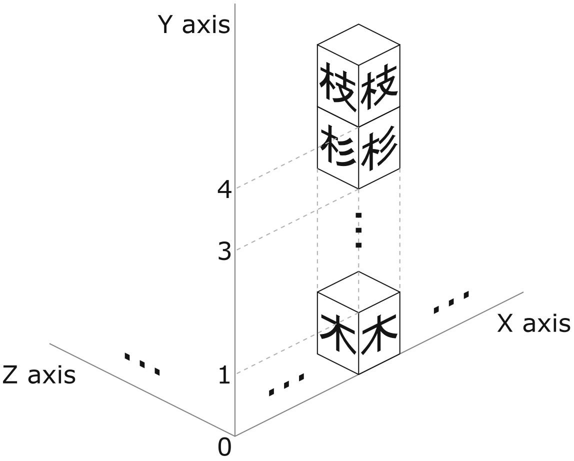


Fig. 1. Illustrating the code structure with the characters 杉, 枝 both of radical 木 and of stroke numbers 3 and 4, respectively

The following two code properties can thus be deduced from the above definitions.

Property 1. Any coordinate $(i, 0, 0)$ $(0 \leq i \leq 213)$ of the code $\mathcal{C}$ designates a radical, and conversely.

Property 2. A radical of index r may also have variants, thus spanning the $(r, 0, z)$ $(0 \leq z)$ line.

Obviously, there may exist several characters of a same radical that have the same number of strokes. To address this multiplicity issue, that is with characters possibly colliding on the Y axis, we rely on decimals: the Y axis thus represents the set of the non-negative rational numbers $\mathbb{Q}^{\geq}$. Even though it is possible to further regulate (order) the affectation of decimals to characters, this would be at the expense of code flexibility: insertion of new characters into the code would be hampered. Hence, decimals are affected to characters of same radicals and same stroke numbers with the only requirement that two characters must not have the same coordinate, which is easily implemented for instance by incrementing the decimal value. Since there may be more than ten characters of same radical and same stroke number, one single decimal is not enough. Therefore, we fix the number of decimals to 6, which is obviously sufficient given that the overall number of Chinese characters is of the ten thousand order (it might border 100,000, but anyway strictly less than a million). For example, considering the two characters 杉 and 村 which are both of radical 木 and of stroke number 3, their X coordinates are $r(木) = 75$ the index of the radical 木 and their Y coordinates are in the range $[3, 4)$, say for instance 3.000000 and 3.000001, respectively.

A few properties of the proposed code are clarified next. Consider two characters c_1 and c_2 of respective coordinates (x_1, y_1, z_1) and (x_2, y_2, z_2). Assume they are of same X coordinates, that is $x_1 = x_2$. Hence,

- they have the same radical;
- if they are both of Z coordinate 0 (i.e., $z_1 = z_2 = 0$), then they are sorted in ascending partial order according to their stroke numbers (i.e., $y_1 \leq y_2$ if and only if c_1 has a stroke number greater than or equal to that of c_2);
- if they both have positive Z coordinates (i.e., $z_1 > 0$, $z_2 > 0$), their respective standard variants (i.e., $(x_1, y_1, 0)$ and $(x_2, y_2, 0)$) are sorted in ascending partial order according to their stroke numbers (i.e., $y_1 \leq y_2$ if and only if the character at $(x_1, y_1, 0)$ has a stroke number greater than or equal to that of the character at $(x_2, y_2, 0)$). In other words, only the characters of Z coordinates 0 are sorted according to their stroke numbers (i.e., according to their Y coordinates).

An illustration in the YZ plane, thus considering characters of one particular radical, is given in Table 1. In this table, "char", "var" and "rad" stand respectively for "character", "variant" and "radical", and the arrow next to the 0 on the Z axis indicates that rows are sorted in ascending order according to the values of this column. Also, by noticing the decimals on the labels of the Y axis as presented previously, one can understand that the characters char2, char3 and char4 have the same stroke number, which is strictly greater than that of char1 and strictly smaller than that of char5.

Table 1. Code structure example in the YZ plane (i.e., the character radical is fixed).

Y axis					
3	char5				
2.000002	char4				
2.000001	char3	char3 var1			
2.000000	char2				
1	char1	char1 var1	char1 var2		
0	radical	rad var1			
	0 ↑	1	2	3	*Z axis*

As a result, we can define a basic lookup function f which partially maps a character to a coordinate in the previously defined three-dimensional space. This function is used to easily locate one character inside the code (i.e., lookup operation). The function f is said partial as it is surjective. As a prerequisite, we define the stroke number function $s : \mathbb{J} \rightarrow \mathbb{N}^*$ which associates to a character its number of strokes (not counting the radical ones). Remark: $\forall c \in \mathbb{J} \setminus \hat{\mathbb{J}}, s(c) > 0$, that is, unless a character is a radical, it has at least one stroke in addition to the those of its radical. Also, let $k : \mathbb{J} \rightarrow \hat{\mathbb{J}}$ be the function that associates a character to its radical.

The function f takes one character or radical as parameter; the domain of the function f is thus $\mathbb{J} \cup \hat{\mathbb{J}}$. The codomain of the function f is $\mathbb{N} \times \mathbb{N}$, thus representing two-dimensional coordinates (x, y), with x, y being non-negative integers. Hence, the function f is used to point at an approximate location in the code, with the $[y, y + 1)$ interval of rational numbers and the Z axis being the approximation range. From this discussion, the basic partial lookup function f is simply defined as $f(c) = (r(k(c)), s(c))$.

Finally, the code also features pointers: effectively, as two characters may have one common variant, pointers are used to avoid duplicates in the code. For example, the radicals 邑 and 阜 both have 阝 as variant. So, assuming that the characters 邑 and 阜 have coordinates $(x_1, y_1, 0)$ and $(x_2, y_2, 0)$, respectively, the coordinate $(x_2, y_2, 1)$ designates a pointer which in turn designates the character 阝 of coordinates $(x_1, y_1, 1)$. Conversely, $(x_1, y_1, 1)$ could designate a pointer which in turn designates the character 阝 of coordinates $(x_2, y_2, 1)$. This situation is illustrated in Fig. 2. A pointer may point at another pointer, in which case the character glyph in the code would be obtained by following the chain of pointers until reaching a non-pointer code element.

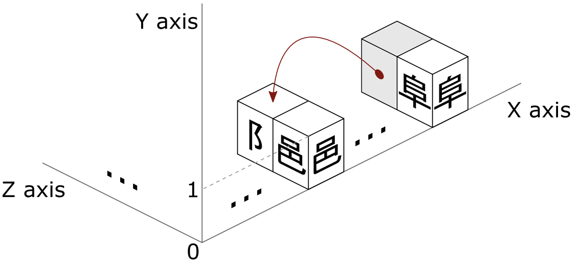


Fig. 2. Illustrating pointers in the code: the radicals 邑 and 阜 both have 阝 as variant; the greyed character block is a pointer

3.2 Refined Code and Enhanced Lookup Function

Obviously, the more the character properties considered, the more accurate the partial lookup function, in other words, steering the initial lookup function from surjection towards bijection. In this section, we propose to refine the proposed code, while retaining its structure and properties, and its basic lookup function described in the previous section.

The improvement relates to the Y coordinate of a character. Previously, it was calculated by taking into account the sole number of strokes of the character, thus almost always resulting in the use of decimals so as to distinguish characters of same radicals and same stroke numbers. In addition to the stroke number, we now consider the stroke order, that is the order in which the strokes that make the character are drawn, and also the types of the used strokes. The former character property (stroke order) is non-ambiguously set by the Japanese government: a character has one unique stroke order. The latter property (stroke types) may be subject to discussion; for the sake of clarity, we restrict the considered stroke types to the basic eight ones as defined for example by Coulmas [12]; they are ㇔, ㇐, ㇑, ㇒, ㇏, ㇀, ㇙ (assimilated with 亅, ㇗, and ㇂) and ㇕ (assimilated with ㇖).

The main idea to implement these two additional character properties into the code without disturbing the code structure is to further rely on decimals. We proceed as follows.

First, it should be noted that the highest number of strokes for a character in Japanese is 84: this is the character mentioned in introduction. Also, we shall consider only the eight basic character strokes as recalled above. Hence, we can represent at the same time the stroke order and the stroke types of a character by using 84 decimals, say from right to left for the stroke order, with each decimal being in the range 1 to 8 to distinguish between the eight stroke types; the eight strokes are numbered from 1 to 8 in the order they are given above. So, for example, the 84 decimals 00...028328 correspond to the character 司. Effectively, this character is drawn in the order ㇕, ㇐, ㇑, ㇕, ㇐. One should note that the radical strokes are included in the decimals.

To these 84 decimals, it is required to further add say 6 decimals so as to distinguish two characters in the exceptional event that they both have the same radical, stroke number, stroke order and stroke types. The characters 丁 and 丅 are such an exceptional character pair, with thus the 84 decimals not enough to distinguish them: they both induce the 00...032 decimals. So, 6 decimals are once again used since this guarantees the possibility of encoding all characters. The 6 decimals are set on the right of the previous 84 decimals so as to retain the code structure previously defined. So, in total, 90 decimals are required in this code enhancement.

This way, as stated at the beginning of this section, the Y coordinate of a character has been refined while retaining the previously described code structure: characters are still ordered on the Y axis according to their stroke numbers. In addition, they are ordered according to their stroke orders and stroke types with the corresponding decimals.

The enhanced lookup function f' can thus be derived as follows. The domain of f' remains unchanged, it is that of f, that is $\mathbb{J} \cup \hat{\mathbb{J}}$. To the difference of f, the codomain of f' is $\mathbb{N} \times \mathbb{Q}^{\geq}$, thus representing two-dimensional coordinates with the Y axis spanning the non-negative rational numbers. Define S the set of the eight basic character strokes as listed previously. Given a character c, assume that $S_c = \{s_0^c, s_1^c, \ldots, s_n^c\} \subseteq S$ is the totally ordered multiset of stroke types such that the character c consists of the $n+1$ strokes s_i^c $(0 \leq i \leq n)$, strokes which are drawn in the order $s_0^c, s_1^c, \ldots, s_n^c$. Let $t : S \rightarrow \{1, 2, 3, 4, 5, 6, 7, 8\}$ be the function that associates a stroke type to its numerical representation (i.e., an integer in the range 1 to 8). Hence, the enhanced partial lookup function f' is defined as

$$f'(c) = \left(r(k(c)), s(c) + 10^{n-83} \sum_{i=0}^{n} 10^i s_i^c \right)$$

As with the basic lookup function f, the 6 additional decimals for collision handling are not covered by the function.

Because there still exist some extremely rare cases where two characters have the same radical, stroke number, stroke order and stroke types – it was indeed hard for the author to exhibit one such character pair as example –, the refined lookup function remains surjective. Hence, even though in most cases the function will directly point at the character being looked up, there may be some possibility that it does not, in which case the function is pointing at a restricted area in the code (actually a very restricted area now that the lookup function has been refined).

In other terms, we have defined a hash function for any Chinese character as used in Japanese. This hash function is a non-perfect one as there remains a few (extremely) rare cases where two distinct characters are hashed to the same value. But because of the extreme rarity of such colliding characters, and obviously of the extreme morphological (rendering) similarity of the two characters, this hash function can be safely used for the conventional hashing purposes: database, cryptography, etc.

4 Database Realisation and Code Visualisation

An important objective of the proposed encoding is improved accessibility. To demonstrate this feature, we practically implement a large character database and illustrate the previously defined code by means of visualisation. As the introduced code is based on a three-dimensional structure, we naturally rely on 3D graphics to illustrate this spatial characteristic of the database.

The data used to realise the proposed database (code) implementation and visualisation has been assembled by the Japanese governmental Information-technology Promotion Agency (IPA); this is the *mojikiban* (文字情報基盤) character database [6], including nearly 70,000 entries. Our implementation includes precisely 56,875 characters of the IPA database. Because the IPA database does not include the stroke order and stroke type information, the code enhancements as proposed in Sect. 3.2 are not implemented here. Completing the IPA database, and thus the resulting code implementation, with such information is part of future works. Finally, it is mentioned as an implementation note that when several radicals were given for one character, the first one was retained (refer for instance to [8] for more on this issue. Besides, this radical plurality explains the low number of characters in the database for a few radicals, such as 匚).

The database realised from the proposed code structure and the corresponding visualisation system were implemented on an Intel i7-6700 CPU, 16 GB RAM machine running a 64-bit Windows 10 operating system. The database was built in two steps as described below. First, selected character information such as

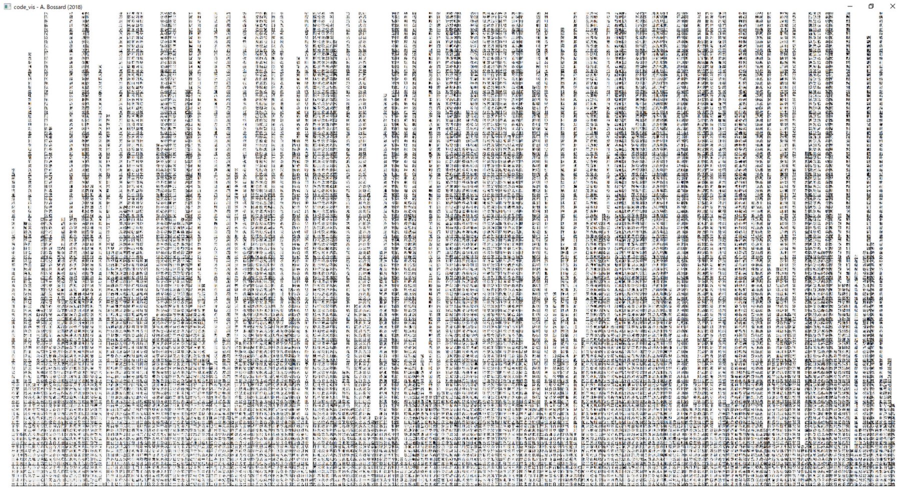


Fig. 3. Overview in the XY-plane of the assembled database (only cut at the top), from the first radical (left) to the last one (right)

radical and stroke number was retrieved from the IPA database and compiled into a flat text file for fast database loading. This process takes about 2 hours 30 min on this experiment machine since requiring multiple IPA database traversals. Importantly, this first step is executed only once: the IPA database is not used afterwards.

Second, the resulting file is loaded into memory and the proposed database structure realised in two passes (in order to address the memory allocation issues). This process is completed in the order of one minute, which is very fast considering the number of characters involved – this loading time is to be compared with, for instance, the time required for the first step. The program takes about 260 MB of memory at run-time, mostly due to distinct textures being used for each glyph. Thanks to this approach, the visualisation (rendering) of the assembled database remains smooth at all time, enabling seamless navigation (e.g., zoom, camera movement) on portions of the rendered code, thus providing a high accessibility for the code as glyphs can be easily located as previously explained.

Several sample illustrations are given below. First, an overview in the XY-plane of the assembled database is shown in Fig. 3. Then, a more detailed view showing characters in a readable form is given in Fig. 4. Next, a detailed view on characters on the Z axis, that is non-regular character variants, are shown in the "rear" view given in Fig. 5.

Finally, the repartition of the regular characters (i.e., the characters of coordinate $z = 0$) considering radicals, and that of the non-regular characters (i.e., the characters of coordinate $z > 0$) are given for reference in Figs. 6 and 7, respectively. These repartitions show character counts for each of the 214 radicals.

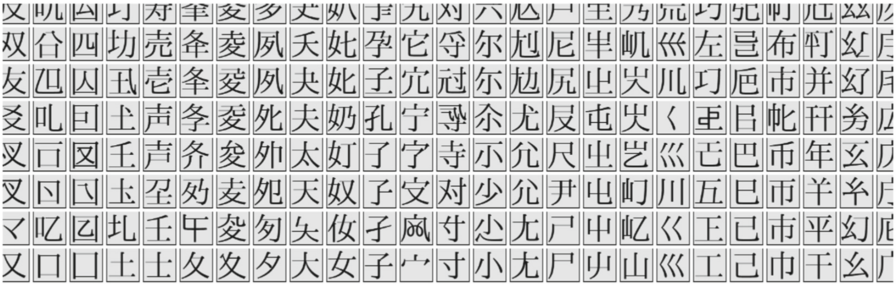


Fig. 4. Zoom-in on an excerpt of the assembled database

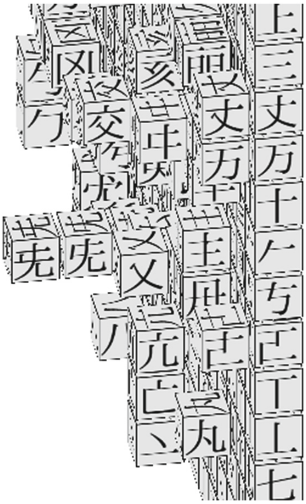


Fig. 5. A detailed view on characters on the Z axis (i.e., non-regular character variants)

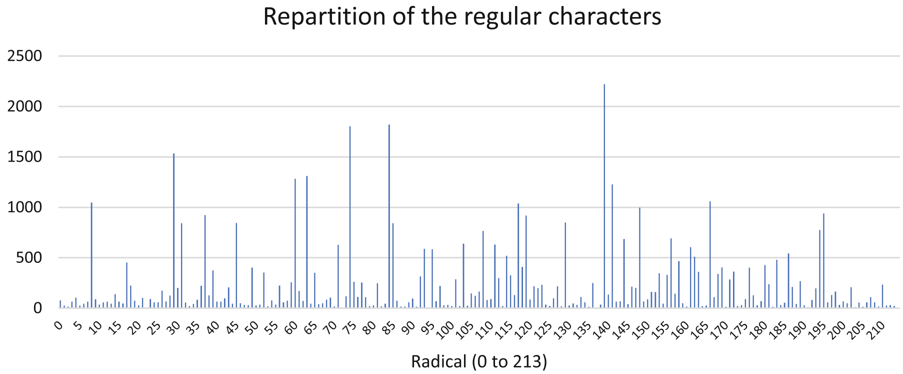


Fig. 6. Repartition of the regular characters (i.e., of coordinate $z = 0$) considering radicals

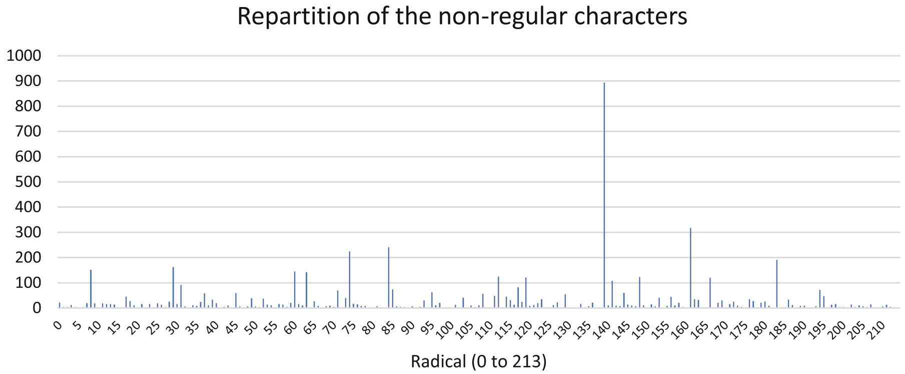


Fig. 7. Repartition of the non-regular characters (i.e., of coordinate $z > 0$) considering radicals

It is especially interesting to see with Fig. 7 that the repartition of non-regular characters is not uniform, most notably with the radical 艸 including in total 894 non-regular characters.

5 Conclusions

We have proposed in this paper a formal database structure for Chinese characters in the form of a character encoding. Unlike previous works such as Unicode and the IPA *mojikiban* database, the described model has been designed to rely on, and retain as much relationship information between entries (i.e., characters) as possible. This makes the proposed code significantly easier to use, for instance for searching operations. Also, the proposed encoding remains flexible by allowing the addition of new glyphs when necessary, and this without disturbing the code (i.e., modifying the code mapping). The number of characters is not limited as in other encodings. In addition to the formal definition of this encoding, as a proof of concept, we have shown how to concretely build such a database, providing a three-dimensional visualisation of the code structure so as to illustrate the spatial characteristic of the proposed database and the induced high accessibility.

Regarding future works, the inclusion of stroke order and stroke type information in the assembled database is meaningful in order to implement the enhanced lookup function. In addition, the optimization of the code for binary representation is definitely relevant given the current architecture of computer systems. And, for instance, the fact that there are in total eight stroke types is interesting from a binary point of view since three bits represent the exact same amount of information, three bits thus sufficing to encode the type of one character stroke.

Acknowledgements. The authors sincerely thank the reviewers for their insightful comments and suggestions. This research project is partly supported by The Telecommunications Advancement Foundation (Tokyo, Japan).

References

1. The Unicode Consortium: The Unicode Standard 5.0. Addison-Wesley, Boston (2007)
2. (Fujitsu) Sekiguchi, M.: 標準化教育プログラム – 第12章 文字コード標準 (in Japanese). Japanese Standards Association (JSA) (2006)
3. Lunde, K.: CJKV Information Processing. O'Reilly Media, Sebastopol (2009)
4. Japanese Industrial Standards Committee (JISC): 7-bit and 8-bit Coded Character Sets for Information Interchange (7ビット及び8ビットの情報交換用符号化文字集合, in Japanese) (1969)
5. Choi, U., Chon, K., Park, H.: Korean Character Encoding for Internet Messages (Request for Comments #1557, Network Working Group). Internet Engineering Task Force, Fremont (1993). https://tools.ietf.org/html/rfc1557. Accessed Mar 2018
6. Information-technology Promotion Agency (Japan): Mojikiban Database (文字情報基盤 文字情報一覧表, in Japanese) (2016). http://mojikiban.ipa.go.jp/. Accessed Feb 2018
7. Morohashi, T.: Daikanwa Jiten (大漢和辞典, in index). Taishukan Publishing, Tokyo (2007)
8. Bossard, A.: Chinese Characters, Deciphered. Kanagawa University Press, Yokohama (2018)
9. Bossard, A., Kaneko, K.: Chinese characters ontology and induced distance metrics. Int. J. Comput. Appl. **23**(4), 223–231 (2016)
10. Ogawa, T., Nishida, T., Akatsuka, K. (eds.): Kadokawa Shinjigen (角川 新字源, in Japanese), revised version. Kadokawa, Tokyo (1994)
11. The Agency for Cultural Affairs, Japanese Ministry of Education, Culture, Sports, Science and Technology (MEXT): Table of the Regular-use Kanji Characters (常用漢字表, in Japanese) (2010)
12. Coulmas, F.: The Writing Systems of the World. Basil Blackwell, Oxford (1989)

Facilitation of Health Professionals Responsible Autonomy with Easy-to-Use Hospital Data Querying Language

Edgars Rencis[1(✉)], Juris Barzdins[2], Mikus Grasmanis[1], and Agris Sostaks[1]

[1] Institute of Mathematics and Computer Science, Faculty of Computing, University of Latvia,
29 Raina blvd., Riga 1459, Latvia
{edgars.rencis,mikus.grasmanis,agris.sostaks}@lumii.lv
[2] Centre of Health Management and Informatics, Faculty of Medicine,
University of Latvia, 19 Raina blvd., Riga 1586, Latvia
juris.barzdins@lu.lv

Abstract. Support for the development of responsible autonomy as opposite to management that is based on direct control is found to be by far more effective approach in healthcare management, especially when it concerns physicians as the most influential group of health professionals. It is therefore important to obtain a process-oriented knowledge system where physicians would be able to autonomously answer questions which are outside the scope of pre-made direct control reports. However, the ad-hoc data querying process is slow and error-prone due to inability of health professionals to access data directly without involving IT experts. The problem lies in the complexity of means used to query data. We propose a new natural language- and semistar ontology-based ad-hoc data querying approach which reduces the steep learning curve required to be able to query data. The proposed approach would significantly decrease the time needed to master the ad-hoc data querying thus allowing health professionals an independent exploration of the data.

Keywords: Responsible autonomy · Hospital management
Self-service knowledge system · Ad-hoc querying · Semistar ontologies
Controlled natural language · Hierarchical data · Medical data

1 Motivation for Further Development of Responsible Autonomy of Healthcare Professionals

Responsible autonomy as opposite to direct control is an organizational system that is based on individual or group autonomy [1]. In healthcare domain, professional autonomy of physicians for independent bedside decisions (accordingly to their specialization certificate) is nowadays increasingly linked also to responsibility to balance clinical and resource dimensions of the whole organization [2]. This is a profound change in the medical profession, and it arises from a variety of interconnected factors linked to one main factor – the enormous progress of medical science that comes at a significant financial cost.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 202–215, 2018.
https://doi.org/10.1007/978-3-319-97571-9_17

The buy-in of the most critical group in healthcare, the physicians, is regarded as mandatory precondition for almost any organizational innovation including also any managerial effort to guide operational and systemic changes by exponentially growing the amount of clinical process efficiency- and quality-oriented healthcare data. In the context of fundamental transformation needed to improve the safety and quality of healthcare, Braithwaite et al. [3] propose "harnessing the natural properties which emerge (often spontaneously) at the interface between the socio (human behavioral) and technical components of complex systems" arguing that "a bottom-up strategy led by clinicians is badly needed to balance the predominantly top-down approaches which frequently result in only modest improvements which are difficult to sustain" [3]. In a wider meaning of the responsible autonomy a bottom-up approach is also needed to access and use the administrative data representing the utilization of limited resources and patient outcomes beyond a single episode of care. However, the complexity and heterogeneity of such data demand either advanced programming and data processing skills to query them, or subordinated data professionals. Both of those options are currently available to management teams, but not to physicians. This fact could be regarded as an obstacle for further development of responsible autonomy, as the care processes are seen through the perspective of performance indicators pre-made by the management rather than through creative exploration of data by physicians themselves.

This situation highlights the need for health professionals to also benefit from advances in IT and to overcome the insufficient opportunity of direct learning from their own practice. To take the responsibility for their own performance and for outcomes of their own concrete decisions, there is a clear demand for a tool that would allow motivated physicians to independently find answers to questions which are outside the scope of pre-made reports.

This paper presents a possible solution to the problem mentioned above. Section 2 gives insight into the related work done in the field of querying the data by domain experts. A lot of work has been done here, but a fully satisfying solution still doesn't exist there. Section 3 introduces the concept of Semistar data ontology as a convenient format for storing healthcare data. Section 4 displays the proposed natural language-based querying language that works upon semistar ontologies. The language is already implemented, and some tests have been done with it. Practical experiments with the language have shown the need for another important facet – the subclass definition feature. This has been described in Sect. 5. Section 6 describes a case study and experiments with groups of domain experts trying to exploit the language in their everyday life. Section 7 concludes the paper.

2 Ad-Hoc Data Querying for Medical Professionals

Direct access to medical data by healthcare professionals would be beneficiary for independently answering questions which are outside the scope of pre-made reports. However, there are no satisfying ad-hoc data querying tools available for end-users which are non-programmers. The main obstacle is the fact that end-users do not have the required skills to define queries by themselves. The main reason is the complexity of the query languages used to get answers from databases. Three main problems are: how to describe data to be easily perceived by end-users; how to formulate queries simply enough for end-users; how to execute queries efficiently in order to retrieve answers in a reasonable time.

Most of the data are nowadays stored in relational databases. The SQL (SEQUEL) language is the de facto standard of querying such data. However, the way data (ER-model) and queries are described in the SQL is too complicated for end-users. There are similar languages for data querying in other types of data storage, e.g. SPARQL for RDF ontologies. The SPARQL as well as SQL requires an accurate formulation of the query (both semantics and syntax) and solid skills in the underlying technology. It thus makes a definition of queries too complex to learn and use. To reduce the complexity wrappers have been made for SQL and SPARQL, e.g. a graphical query builder "Graphical Query Designer" for SQL Server, a graphical query builder "ViziQuer" [4] for SPARQL and RDF databases, a form-based tool which uses standard GUI elements (e.g. tables and lists) and wizards "SAP Quick Viewer SQVI". Another means for direct data access is Self-Service Business Intelligence by Microsoft [5]. It offers a set of tools called "Power BI". Power BI allows an end-user creating advanced visualizations of data and performing the data analysis through a spreadsheet application (MS Excel). Although even IBM with its Watson Analytics [6] is dealing with the direct data access problem, a flawless solution has not been found yet. The most significant downside of these approaches is a steep learning curve necessary to master a new query approach and to understand the way the data are described.

Another feasible option for medical professionals is a natural language or more precisely – a natural language interface to databases (NLIDB) through which end-users can query data in the relational databases [7–11]. However, the definition of the accurate query itself is a hard task for end-users without IT background. NLIDB systems are not easily usable by end-users due to the problem of linguistic coverage. Firstly, explanations to users are needed about what the database contains and what types of questions can be asked. Data descriptions used by database experts (like ER models) are too complicated and contain too many technical details to be useable for the understanding of the data. Secondly, the software does not properly understand what a user means by a query due to the ambiguity and richness of the natural language. Data schema representation satisfying requirements of both human and software is needed. NLDIB uses ontologies to define concepts, relationships between them and their properties. Concepts form a vocabulary that can be used by end-users. Although ontologies hide technical details they require the knowledge of basic object-oriented modeling principles. The NLIDB approaches have thus not been used widely, especially for complex querying with nontrivial calculations.

Process visualization as "an easy way to discover and create value" [12] is often mentioned in engineering and lean management literature as another way to explain the data to the end-user. However, meaningful description and visualization of pathways of individual patients as a clinical process of the whole hospital is a complicated task due to its extremely variable nature. Therefore Vos et al. [12] propose to use a generalized visual model of hospital processes for clinicians to understand what data, where and when during the treatment are recorded. If the clinicians understand the localization of the data in the model they may explicitly and clearly formulate explorative questions for building process-oriented knowledge base. As examples of the representative clinical process variables or of numerical indicators (with desirable limits set) the following should be mentioned: average treatment time, mortality rate and other clinical outcomes of patients, frequency of hospitalization, waiting time at various stages of the process, patient satisfaction, utilization of resources (rooms, equipment, staff), costs of certain manipulations or patient groups, costs of certain patients, details of particular event – when it has happened, how many times, within what time interval etc.

3 In Search of the Appropriate Format for Storing Healthcare Data

When we think about the various data representation formats from which to choose, several options can be available. One of the most exploited data storage formats is the relational database, because usually the data can indeed be represented in the form of ER model. If we choose to use this data storage format, we are later able to query the ontology using the SQL as a query language. This is a common solution, and thus it is quite easily to implement it.

However, we must consider not only the ease of implementation of the chosen solution, but also its friendliness to end-users that are not IT specialists. As was stated in Sect. 1, the main types of users of our system will be healthcare professionals (managers and physicians), so we cannot assume that for them the ER model would be the best representation of healthcare data in a natural and understandable way. Since ER model is almost never granular (naturally dividable into data slices), it is usually not easily understandable by non-programmers [13]. Moreover, although the SQL language was initially designed to be used by standard end-users, hardly any non-programmer has nowadays acquired the necessary skills to be able to understand SQL queries, not to mention writing them him/herself.

We therefore have to cope with at least two challenges: (1) how to depict the data ontology to be easily understandable by healthcare professionals; (2) how to develop a query language based on this representation of the underlying data ontology, such that a healthcare professional could formulate queries him/herself and understand their answers. Finally, if we had developed such a user-friendly query language, we would then encounter also the third challenge: how to implement the query language efficiently enough in order to get the answer to a sufficiently wide class of queries in a reasonable time. These three challenges together form the so-called “3How” problem which we have described in more detail in our previous work [13].

If we now think about the most suitable format for storing healthcare data, we should look at how these data were stored before they were digitalized. When the information about patients was filled in by hand, hospitals used so-called patient cards where each patient had his separate card and each card contained information about each occurrence of this patient in the hospital, and each occurrence contained information about the treatments provided for the patient in this particular occurrence and so on. This division into smaller and smaller subdivisions is a very natural way of storing healthcare data, and it is also very familiar to healthcare professionals. We have therefore chosen exactly such structure to be the basis of the data ontology that solves the first challenge of the abovementioned “3How” problem. The described structure is known in literature as the reversed star data schema, because in it “certain key characteristics of the classic star schema are ‘reversed’” [14]. Indeed, in typical situations we always have one central class (the class “Patient” in this case), from which several paths can lead to other connected classes that have the relation one-to-many, as can be seen in Fig. 1. It is also known that any database organized in the third normal form can be converted to a reversed star schema thus making the reversed star ontologies very powerful [15]. In addition to the classical reversed star ontologies we also allow addition classes outside the star which are devoted for registers and classifiers (classes “CPhysician”, “CDiagnosis” and “CManipulation” in Fig. 1). We call such enriched ontologies the semistar ontologies, and we have described them in detail in our previous work [13, 16–19]. The simplified version of a semistar ontology seen in Fig. 1 has been introduced for use in Riga Children’s Clinical University Hospital (RCCUH).

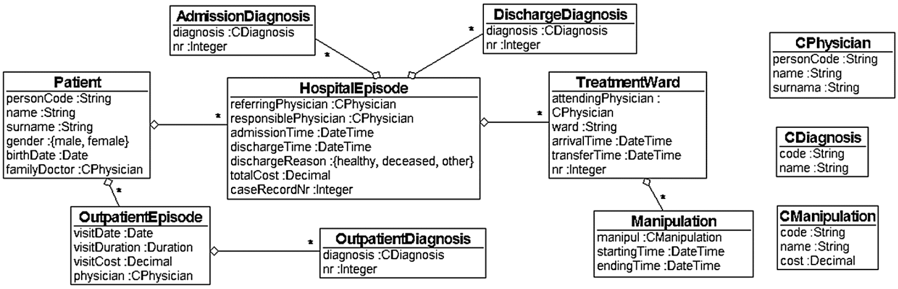


Fig. 1. Semistar ontology exploited in Riga Children’s Clinical University Hospital

Semistar ontologies are by their nature granular, that is – they can be naturally divided into slices [13, 19] where each slice contains concluded information about one particular patient. This intrinsic feature of semistar ontologies has allowed us developing a new kind of querying language that would solve the second challenge of the abovementioned “3How” problem. The goal for the language was to be very easily understandable by healthcare professionals, so that they would be able to formulate their own ad-hoc queries in a convenient manner. This basic feature of the language was tested in an experiment with healthcare professionals. This is described in more detail in Sect. 6. The language itself is briefly described in Sect. 4.

Finally, we have also solved the last challenge of the “3How” problem by developing a very efficient implementation of the proposed query language [13]. Details of the implementation are, however, outside the scope of this paper.

4 Development of a Natural Language-Based Query Language

The main reason for choosing a semistar ontology for storing the data is its simplicity from the end-user point of view. It has been proved that this format is inherently intuitive for domain experts [17, 18]. It can be explained by the fact that this format is very similar to the old paper forms that all healthcare professionals are familiar with. Moreover, the data storing format of the semistar ontology also allows us developing an easy-understandable query language in a natural manner. The query languages are usually complicated due to the technical details forced by non-intuitive data storage formats (such as relation databases and their respective query language SQL). For example, the concept of a role name can be quite hard to comprehend by non-programmers. In the case of semistar ontologies we have managed to not embed such technical details in the query language. As a result we have succeeded in developing a query language that is based purely on the natural language and exploits only those concepts included in the ontology. Therefore during the language development process we had hoped that also the query language would turn out to be understandable by end-users who are familiar with its underlying semistar ontology and its concepts. Our hopes proved right after we performed the end-user survey described in Sect. 6.

Of course, the natural language, on which the query language is based, is still quite controlled. Nevertheless, it is much more suitable for healthcare professionals (physicians and hospital managers) than SQL-like languages.

The level of control of the natural language grammar forms allowed in our query language is defined by seven sentence constructs called sentence templates. These templates are described in more detail in our previous work [17], but we are listing them also here for the purpose of gaining more general understanding of the proposed query language.

The sentence templates are written in a natural language and contain placeholders for inserting conditions and expressions written in special syntax. Formal explanation of this syntax is not a part of this paper, but it can be done using a special kind or grammar called the Definite Clause Grammar. Here we will instead explain the construction of those conditions using examples.

Let us now introduce the seven sentence templates illustrated by examples.

T1. COUNT AClass [x] WHERE <selection condition>

Semantics: counts instances of AClass, which satisfy the selection condition.

Example:

- COUNT Patients, WHERE EXISTS HospitalEpisode, WHERE referringPhysician=familyDoctor (count of patients who have been referred to hospital by their family doctors);

T2. {SUM/MAX/MIN/AVG/MOST} <attribute expression> FROM AClass [x] WHERE <selection condition>

Semantics: selects instances of AClass, which satisfy the selection condition, calculates the attribute expression for each of these instances obtaining a list to which the specified aggregate function is then applied. Example:

- SUM totalCost FROM HospitalEpisodes, WHERE dischargeReason=healthy AND birthDate.year() =2012 (how much successful treatments of patients born in 2012 have cost);

T3. SELECT FROM AClass [x] WHERE <selection condition> ATTRIBUTE <attribute expression> ALL DISTINCT VALUES

The semantics is obvious. Example:

- SELECT FROM HospitalEpisodes, WHERE dischargeReason=deceased, ATTRIBUTE responsiblePhysician.surname ALL DISTINCT VALUES;

T4. SHOW [n/ALL] AClass WHERE <selection condition>

The semantics: shows n or all instances of AClass which satisfy the selection condition.

T5. FULLSHOW [n/all] AClass WHERE <selection condition>

The semantics: the same as "show", but shows also the child class instances attached to the selected AClass instances.

T6. SELECT AClass x WHERE <selection condition>, DEFINE TABLE <x-expr'1> [(COLUMN C1], …, <x-expr'n> [(COLUMN Cn)] [, KEEP ROWS WHERE <Ci selection condition>] [, SORT [ASCENDING/DESCENDING] BY COLUMN Ci] [, LEAVE [FIRST/LAST] n ROWS]

The semantics: selects all instances of AClass, which satisfy the selection condition, then makes a table with columns C1 to Cn, which for every selected AClass instance x contains an individual row, which in column C1 contains the value of the <x-expr'1>, …, in column Cn contains the value of the <x-expr'n>. Then it is possible to perform some basic operations with the table like filtering out unnecessary rows, sorting the rows by values of some column and then taking just the first or the last n rows from the table. Example:

- SELECT HospitalEpisodes x, WHERE dischargeReason=deceased, DEFINE TABLE x.surname (COLUMN Surname), x.dischargeTime.date() (COLUMN Death_date), (COUNT x.Manipulation, WHERE manipul.code=02078) (COLUMN Count_02078), (SUM manipul.cost FROM x.Manipulation, WHERE manipul.code=02078) (COLUMN cost_02078);

T7. There are two more cases in the definition of the table, where table rows come from some other source, not being instances of some class. Being very similar, these two cases form two subtemplates of the last template:

(a) SELECT FROM AClass [a] WHERE <selection condition> ATTRIBUTE <attribute expression> ALL DISTINCT VALUES x, DEFINE TABLE…
(b) SELECT FROM INTERVAL (start-end) ALL VALUES x, DEFINE TABLE…

The semantics of both cases is obvious. Examples:

- SELECT FROM TreatmentWards ATTRIBUTE ward ALL DISTINCT VALUES x, DEFINE TABLE x (COLUMN Ward), (SUM manipul.cost FROM Manipulations, WHERE ward=x) (COLUMN Cost);

- SELECT FROM INTERVAL (1–12) ALL DISTINCT VALUES x, DEFINE TABLE x (COLUMN Month), (COUNT HospitalEpisodes, WHERE admissionTime.month()=x) (COLUMN Episode_count) (MOST diagnosis.code FROM AdmissionDiagnoses, WHERE nr=1 AND admissionTime.month()=x) (COLUMN Most_frequent_main_diagnosis).

5 The Subclass Definition Feature

Practical experiments with former version of our query language had shown [17] that there was yet one more very important feature that should be added to increase the usability of the language – the subclass definition feature. It was very common that users wanted to exploit a certain subclass of some ontology class multiple times in their querying process. It was not very convenient to repeat the part of the query defining the class each time that class is needed. It would be much more convenient if one could develop the class only once and then use the class name in each of the subsequent occurrences.

Let us inspect an example of when the introduction of a subclass could be beneficial. Let us say we want to deal with patients who have been healed in the hospital and calculate the total cost of their hospital episodes in which they have been healed. The query that answers this question looks like this:

SUM totalCost FROM HospitalEpisode WHERE dischargeReason = healthy

Now we would perhaps like to refine our query and ask how many of those episodes lasted more than a week:

COUNT HospitalEpisode WHERE dischargeReason=healthy AND discharge-Time-admissionTime>7d

Or we could want to create a table of 10 most expensive episodes and show the responsible physician for each of these episodes:

SELECT HospitalEpisode WHERE dischargeReason=healthy, DEFINE TABLE totalCost (COLUMN Cost), name (COLUMN Patient name), surname (COLUMN Patient surname), responsiblePhysician.name (COLUMN Physician name), responsiblePhysician.surname (COLUMN Physician surname), SORT DESCENDING BY COLUMN Cost, LEAVE FIRST 10 ROWS

We can see that here persists the need for repeating the definition of hospital episodes with a healthy outcome. It would be more convenient if we had a class name (e.g. HealthyEpisode) for such episodes. When introducing new features in the language we must consider two things – the usability from the end-user point of view and the performance of query execution from the implementation point of view. To optimize the performance we had to consider several possible subclass implementation options:

(1) to translate each definition of subclass S of class A to a predicate PS<A> and then substitute every occurrence of class S in queries with A and PS<A>;
(2) to create a fully materialized view in the subclass definition moment by calculating values of the defining predicate for each instance of the class A and to store this table in the persistent database;
(3) to create the subclass definition table when the subclass is first used and to store this table in the session memory;
(4) to store the information about the subclass affiliation in the instance level of the class A by using the lazy evaluation principle and to calculate the affiliation of particular instance to the subclass S only when the instance is being used.

Options 2–4 are somewhat similar one to another, because they all propose storing the information about the instance affiliation to the subclass in the memory. Option 1 allows us to save memory by introducing another layer of translation into the query where the occurrences of the subclass are substituted with its defining predicate call before the query is passed to the real translation into the executable Java code. The negative aspect of this option is that it could compromise the performance of the query execution in cases when the subclass defining predicate is very complex (i.e., it includes aggregate functions over the base class neighbors). Option 1 would perhaps be the best solution in case we would restrict the definition of the subclass S only to the attributes of the superclass A. That is, however, often not the case. The subclass definition is usually based on various conditions and may consider also attributes of other classes. Nevertheless, we chose Option 1 as the first attempt to implement the subclass definition feature in our query language, because we wanted to test whether the query execution performance would indeed be compromised and to what degree. In future we are planning to test also other subclass definition feature implementation approaches.

The subclass definition feature consists of two parts – the subclass definition part and the subclass application part. To allow end-users to define subclasses we have introduced another type of sentence templates (let us call it T8) in the query language that looks like this:

T8. DEFINE SClass=AClass [x] WHERE <selection condition>

Semantics: defines SClass as a subclass of AClass whose instances satisfy the selection condition.

After the subclass is defined it is accessible to end-users just like any other class of the ontology. When the subclass is used in the query formulation all its occurrences are wisely substituted with its definition before the query is passed to its translation phase. For example, one can now define the class HealthyEpisode as a subclass of the class HospitalEpisode:

DEFINE HealthyEpisode=HospitalEpisode WHERE dischargeReason=healthy

Now the class HealthyEpisode can be used as a normal class to, for example, calculate the total cost of all such episodes with a healthy outcome that have lasted for more than a week:

COUNT HealthyEpisode WHERE dischargeTime-admissionTime>7d

Such a query would then be automatically translated into a full query that does not exploit the subclass HealthyEpisode:

COUNT HospitalEpisode WHERE dischargeReason=healthy AND dischargeTime-admissionTime>7d

This translation result would then be passed to the translator to Java code and processed normally. Our first observations regarding the performance of such query execution show no major loss in most cases. Of course, the subclass defining predicate is not limited in scope, and it can grow quite big in some cases thus compromising the query execution performance to a greater degree. However, our experiments show that such complex predicates are not typical in everyday work of domain experts of the medical domain. Nevertheless, we must continue to explore the subclass definition feature in more detail to obtain more precise data of its performance.

6 Concept Testing

According to the description of a significantly wide range of information that semistar ontologies could cover it was stated in Sect. 3 that the hospital data scheme and query language are potentially “readable” by the physicians. We therefore wanted to test two aspects of our concept among domain experts – (1) whether the proposed ontology and the language are sufficiently expressive for practical tasks and (2) whether it is simple enough to be used by a motivated physician – a domain expert. To administer such a test we implemented a software prototype which allowed users writing queries in our query language and executing them upon a sample database.

Initially the expressiveness was tested for a simplified hospital data scheme. We wanted to find out whether the simplification with the aim of displaying data scheme on one A4 page in an easy-to-use and memorizable manner would not compromise meaningful data analysis. The capability of data analysis was compared against a set of performance indicators used in hospitals for local operational management and indicators measured nationally to benchmark various international quality and efficiency aspects of patient care. The analysis we performed proved a potential of the elaborated scheme to cover full data set needed for the calculation of all currently used patient- and admission-based indicators.

The expressiveness of the query language was tested when introduced for the preparation of the detailed annual administrative and clinical reports for Intensive Care Unit of Riga Children's Clinical University hospital. The language proved to be sufficiently expressive for this rather complicated task. During two years of testing and further improving, it became a working language for the involved managing staff to formulate data queries without referring to SQL programmers in most of the cases thus approaching the self-service criteria. With or without the help of a limited set of pre-written example queries users formulated various types of queries independently – queries that have a single number result, queries that highlight multiple data items for a single patient with a particular tracer and queries that create data tables with or without some calculations. Regarding calculations and tables our aim was to incorporate only the very basic mathematical and logical calculations in the language, because there is always a possibility to export a table containing all the necessary data to MS Excel or other tools specially designed for statistical analysis.

The last part of concept testing was focused on the practical teaching aspects of the language to potential end-users who were not previously involved in the process of language development or in previous analysis of hospital data in general. We performed both individual and group experiments. The potential of the language to be taught to domain experts was best demonstrated in an experiment with a random group of domain experts – experienced health professionals in M.Sc. in the Nursing studies course at the University of Latvia. Following a regular lecture in the Healthcare management module covering general aspects of data management in healthcare, we presented elaborated simplified hospital data scheme, the querying language and the experimental web-based tool for querying data. Students were also informed that the data they will be offered for analysis, albeit anonymous, are real (our previous experience had showed that domain experts were significantly more motivated to learn querying on real data, because it often allowed satisfying natural curiosity related to previously unknown details and patterns of their everyday work). The first third of our two-hour long lecture was dedicated to the explanation of the simplified data ontology (the semantics of such concepts as 'class', 'attribute', 'classifier', 'cardinality', etc.). The rest of the lecture was devoted to an example-based explanation of the language. We chose the teaching approach based on the principles of natural language acquisition, because we had already found in early stages of elaboration of the language that the technical explanation of the language syntax is too cumbersome and uninspiring for potential end-users who are busy health professionals.

To explain the language we started with very simple examples like asking the number of patients treated at hospital with a following inquiry of the number of treatment episodes. Since there are definitely cases of patients treated at hospital for more than one time in the given year, technically correct queries give different results. In such a way we demonstrated the ambiguity of a natural language and the importance of formulating questions precisely already in the natural language, so that they can be used to construct the query afterwards. The next question regarding the number of patients treated more than once allowed us to introduce the important notion of existence (number of ‘patients’ for whom there ‘exists’ at least one ‘episode’) in an easy manner. Following the introduction of each new construction it was strengthened with a set of related examples. For example, we asked the number of patients having at least one episode with a registered treatment in the particular treatment ward or the number of episodes with the patient being treated in the same ward (additionally counting also cases when the same patient was admitted several times during the given year), or finally the number of treatment cases in the same ward (additionally counting the cases when the patient had changed wards during one episode and was admitted in the particular ward more than once). In a similar way we advanced from basic constructions to more sophisticated ones while introducing other concepts. To mention just few of many examples used, a list follows here of some of the used inquiries: the number of episodes started between 5 PM and the midnight (including); the workload of a particular department (the total duration of all treatment cases in that ward); the number of episodes when the patient was referred to a hospital by relevant family doctor; the number of unique patients referred to a hospital; the number of episodes when the attending physician was the particular person; the number of patients admitted (in fact – the episodes initiated) in the particular day; the number of hospital and outpatient episodes in the particular month; the number of treatment episodes longer than the particular number of days; the number of patients having the defined age range at the time of admission; the number of patients having the treatment costs above the defined limit. The introductory lecture also included several examples used to generate a list of values for some attributes according to the defined selection (e.g. the three most common main diagnoses (codes) for patients younger than 5 years), or an extended list (e.g. the most expensive manipulation performed to patients younger than 5 years showing costs and manipulation codes).

Following the lecture we gave students homework in order to test their level of understanding. In its first part the students were asked to translate the set of pre-written queries into a good natural language. Its second part was the opposite – to rewrite natural language sentences into the formal query language. Students were advised to complete homework as soon as possible. As the result in the group of 15 students there were 9 students who did the reading task and 3 students who also did the writing task of the correctness level of at least 90%. We consider this result inspirational, because it shows that a significant number of health professionals previously not being involved in healthcare data analysis acquired important elements of the proposed querying language relatively easily. This confirms the need for further research in the use and development of a controlled natural language for querying data by domain experts at least within the healthcare domain.

7 Conclusions and Future Work

In this paper we have proposed an easy-to-learn data querying language based on the principles of a natural language. Health professionals admit that clinical governance can only be effective if three components are integrated together – the financial control, the service performance and the clinical quality. In such a case the clinicians would be engaged, thus generating the service improvements [20]. To establish such an engagement it is also very important to understand how the data underlying the clinical process are collected and integrated, as well as what are the meanings of the numerous data elements. Self-regulation ability of health professionals would greatly benefit from the data querying approach proposed in this paper. This is in its turn necessary to optimize both technical and social aspects of the system and to move towards the goal of a more effective hospital.

We believe our proposed query language is much easier to use than various traditional query languages that are based on data located in the databases (like the SQL language). We have tested our approach by performing practical experiments which showed that the domain experts in the healthcare domain are able to learn the language quite rapidly and to use it to extract the necessary information from the available data stored in a form of a semistar ontology.

Another big topic that we plan to address in future is the mechanism of access rights. Data located in ontologies of medical domain are very sensitive and therefore various types of users are allowed to access different parts of these data. We are planning to introduce the notion of a role in the language and to supplement the language with some constructs for defining the subclass of data slices that would be accessible to a particular role.

The query language described in this paper is based on a rather controlled natural language. Our future goals therefore also include reducing the level of control by allowing users to express their needs in more informal manner. That way the query language would become even more usable for healthcare professionals. Our aim here is to allow formulating queries by providing only some basic keywords which our system would then recognize and try to generate the most probable queries in the language proposed in this paper. Since it has been proved in our end-user experiments that our language is very readable, we hope that the user would then be able to choose among the proposed queries and affirm the one he/she has actually meant. This would be the next logical step taken towards even more user-friendly natural language-based query language.

Acknowledgements. This work is supported by the ERDF PostDoc Latvia project Nr. 1.1.1.2/16/I/001 under agreement Nr. 1.1.1.2/VIAA/1/16/218 "User Experience-Based Generation of Ad-hoc Queries From Arbitrary Keywords-Containing Text" and the joint project of University of Latvia and Centre for Disease Prevention and Control "Towards a public monitoring system for the quality and efficiency of health care" under agreement Nr. ZD2017/20443.

References

1. Friedman, A.: Responsible autonomy versus direct control over the labour process. Cap. Cl. **1**(1), 43–57 (1977)
2. Degeling, P., Maxwell, S., Kennedy, J., Coyle, B.: Medicine, management, and modernisation: a "danse macabre"? BMJ Br. Med. J. **326**(7390), 649–652 (2003)
3. Braithwaite, J., Runciman, W.B., Merry, A.F.: Towards safer, better healthcare: harnessing the natural properties of complex sociotechnical systems. Qual. Saf. Health Care **18**(1), 37–41 (2009). https://doi.org/10.1136/qshc.2007.023317
4. Zviedris, M., Barzdins, G.: ViziQuer: a tool to explore and query SPARQL endpoints. In: Antoniou, G., et al. (eds.) ESWC 2011. LNCS, vol. 6644, pp. 441–445. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-21064-8_31
5. Aspin, A.: Self-service business intelligence. In: High Impact Data Visualization with Power View, Power Map, and Power BI, pp. 1–18. Apress, Berkeley (2014)
6. IBM: Watson Analytics (2017). https://www.ibm.com/watson-analytics
7. Androutsopoulos, I., Ritchie, G.D., Thanisch, P.: Natural language interfaces to databases – an introduction. Nat. Lang. Eng. **1**(1), 29–81 (1995). https://doi.org/10.1017/S135132490000005X
8. Li, F., Jagadish, H.V.: Constructing an interactive natural language interface for relational databases. J. Proc. VLDB Endow. **8**(1), 73–84 (2014)
9. Llopis, M., Ferrández, A.: How to make a natural language interface to query databases accessible to everyone: an example. Comput. Stand. Interfaces **35**(5), 470–481 (2013)
10. Papadakis, N., Kefalas, P., Stilianakakis, M.: A tool for access to relational databases in natural language. Expert Syst. Appl. **38**, 7894–7900 (2011)
11. Popescu, A.M., Armanasu, A., Etzioni, O., Ko, D., Yates, A.: Modern natural language interfaces to databases: Composing statistical parsing with semantic tractability. In: Proceedings of the 20th International Conference on Computational Linguistics, COLING 2004, article no. 141 (2004)
12. Vos, L., Chalmers, S.E., Dückers, M.L., Groenewegen, P.P., Wagner, C., van Merode, G.G.: Towards an organisation-wide process-oriented organisation of care: a literature review. Implement. Sci. **6**(1), 8 (2011). https://doi.org/10.1186/1748-5908-6-8
13. Barzdins, J., Rencis, E., Sostaks, A.: Data ontologies and ad hoc queries: a case study. In: Haav, H.M., Kalja, A., Robal, T. (eds.) Proceedings of the 11th International Baltic Conference, Baltic DB&IS, pp. 55–66. TUT Press (2014)
14. Kasprzyk, A., Keefe, D., Smedley, D., et al.: EnsMart: a generic system for fast and flexible access to biological data. Genome Res. **14**, 160–169 (2004)
15. Zhang, J. et al.: BioMart: a data federation framework for large collaborative projects. Database J. Biol. Databases Curation (2011). http://doi.org/10.1093/database/bar038
16. Barzdins, J., Grasmanis, M., Rencis, E., Sostaks, A., Barzdins, J.: Self-service ad-hoc querying using controlled natural language. In: Arnicans, G., Arnicane, V., Borzovs, J., Niedrite, L. (eds.) DB&IS 2016. Communications in Computer and Information Science, vol. 615, pp. 18–34. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-40180-5_2
17. Barzdins, J., Grasmanis, M., Rencis, E., Sostaks, A., Barzdins, J.: Ad-hoc querying of semistar data ontologies using controlled natural language. In: Databases and Information Systems IX. Frontiers in Artificial Intelligence and Applications, vol. 291, pp. 3–16. IOS Press (2016). https://doi.org/10.3233/978-1-61499-714-6-3

18. Barzdins, J., Grasmanis, M., Rencis, E., Sostaks, A., Steinsbekk, A.: Towards a more effective hospital: helping health professionals to learn from their own practice by developing an easy to use clinical processes querying language. Procedia Comput. Sci. J. **100**, 498–506 (2016). https://doi.org/10.1016/j.procs.2016.09.188. International Conference on Health and Social Care Information Systems and Technologies
19. Barzdins, J., Rencis, E., Sostaks, A.: Granular ontologies and graphical in-place querying. In: Short Paper Proceedings of the PoEM, CEUR-WS, vol. 1023, pp. 136–145 (2013)
20. Smith, L.F.P., Shepperd, J.: Making clinical governance work for you. Br. Med. J. **322**(7302), 1608 (2001)

Efficient Model Repository for Web Applications

Sergejs Kozlovičs(✉)

Institute of Mathematics and Computer Science,
University of Latvia, Raina blvd. 29, Riga 1459, Latvia
sergejs.kozlovics@lumii.lv

Abstract. Many model-based applications have been developed with standalone usage in mind. When migrating such applications to the web, we have to think about multiple users competing for limited server resources. In addition, we encounter the need to synchronize models via the network for client-side access. Thus, there is the risk that the model storage could become a bottleneck.

We propose a model repository that deals with these issues by using an efficient encoding of the model that resembles its Kolmogorov complexity. The encoding is suitable for direct sending over the network (with almost no overhead); it can also be used "as-is" in memory-mapped files, thus, utilizing the OS paging mechanism. By adding just 3 automatic indices, all traverse and query operations can be implemented efficiently. Our tests show that the proposed model repository outperforms other repositories concerning both CPU and memory and is able to hold 10,000 and more instances at the same time on a single server.

Keywords: Models · Model repository · Web applications

1 Introduction

In 2018, the growth in internet users has reached 4 billion people constituting over half of the world's population [11]. The wide availability of the internet combined with the development of cloud technologies made it much easier and cheaper to deploy web applications than it was 20 years ago. The obvious benefit of web applications is that they are available right away, without the need to install them. In addition, they are always up-to-date and accessible throughout the globe from different devices and operating systems. That is why many standalone applications are now being moved to the web environment. However, developers of web applications face additional issues such as the presence of network overhead and competition between multiple connected users for limited server resources.

The scope of this paper lies within classical model-based applications that have to be moved to the web. By *model-based application* we mean an application

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 216–230, 2018.
https://doi.org/10.1007/978-3-319-97571-9_18

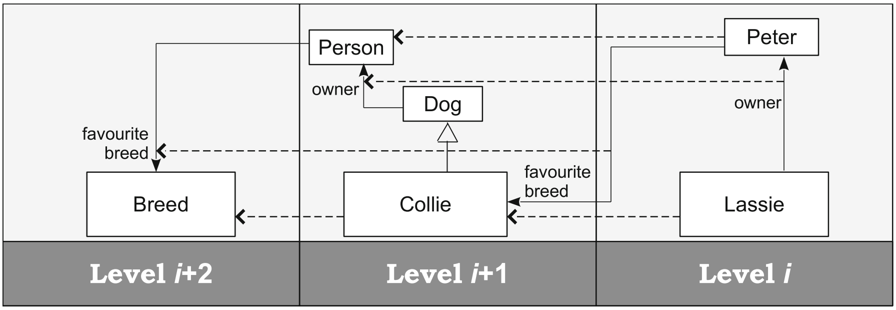


Fig. 1. An example of multiple mixed meta-levels (dashed lines represent the "instance-of" relation). The relation "favourite breed" between Person and Breed as well as the link between Peter and Collie cross two adjacent meta-levels.

that stores data in MOF[1]-like models and processes these data by corresponding model transformations [15,17]. We say "MOF-like" models, since in practice alternative implementations such as Java-based EMF/ECore are used [1,20]. By *model transformations* we mean not only specific programs written in some model transformation language (like MOLA, Lx, Epsilon, ATL, VIATRA, etc.), but also programs written in traditional programming languages (like Java or C++) and that are able to access MOF-like models via some API (e.g., ECore API) [7–9,13,21].

In model-based applications models are saved in a storage that we call *model repository*. This concept differs from the database concept in the following main points:

- the repository does not need to be able to perform complex queries- that is the task of model transformations; the repository just has to implement simple model traversal and update operations;
- in a repository, the model is usually loaded into memory, since transformations use the model intensively for both read and write operations; that differs from databases (not only SQL, but also graph and document databases), which are optimized for performing queries, while update operations are much slower (usually involving re-arranging indices)[2];
- model repository internal structures and APIs are tailored for storing models, i.e., both object-level data (objects, their attributes, and links) as well as meta-data (object types and their properties) can be stored. In some use cases multiple meta-levels are required (see Fig. 1), thus, model repository should be able to store them all. Although existing SQL and no-SQL databases can be tamed for these purposes (e.g., via object-relational mapping, ORM), such approach is more comprehensive and less efficient than using a true model repository directly [5,6].

[1] MOF (Meta-Object Facility) is a standard developed by OMG (Object Management Group) for describing formal models [15].

[2] For example, MongoDB write operations can be up to 10 times slower than read operations.

While migrating model-based applications to the web, there is the risk that the model repository could become a bottleneck, since model transformations use it intensively to implement business logic of model-based applications. Existing model repositories revealed two extremes: either a repository was memory-efficient, but not CPU-efficient (like ECore), or vice-versa (like the "New Repository" JR presented in 2010 [18]). Moreover, the internal encoding of model repositories was usually concealed, thus, sending the whole repository content via the network required to rely on the repository API, which was slower than if we had access to internal data structures. Thus, there remained a need for a fast repository that could be used in the web environment.

In this paper we propose a new model repository that is *both* CPU- and memory-efficient. It is also designed for fast synchronization via the network. Besides, the proposed repository has all the necessary functionality for storing and traversing models at different meta-levels. Our approach relies on a specific encoding of models.

The next section presents our idea. Section 3 reveals some interesting implementation details. In Sect. 4 we provide quantitative test results that confirm the feasibility of our approach. Finally, we discuss the potential of the proposed repository and conclude the paper (Sects. 5–6).

2 The Main Idea

When deciding which API the upcoming model repository has to implement, we aimed for an API that would be compatible with existing repositories. However, we tried to avoid high-level APIs (like *Epsilon Model Connectivity Layer* or *ATL Model Handler Abstraction Layer*), since they are not efficient (e.g., linked objects can only be set as a list, even when we need to include/exclude just one object), they conceal internal data structures too much, and they are hard to use with mixed meta-levels [8,13]. Instead, we focused on a low-level Repository Access API (RAAPI)[3]. RAAPI can be viewed as repository assembler, thus, the sequence of RAAPI calls can be treated as an assembly program for creating the content of the repository from scratch. This resembles how a sequence of low-level Turing machine operations results in the given string. We show below that the sequence of RAAPI operations can be kept short enough, thus, it can be used as an efficient encoding of a model (this resembles the Kolmogorov complexity concept with the difference that the sequence of operations results in a model instead of a string).

An interesting feature of RAAPI is that it was developed with Šostaks' conjecture in mind [14]:

> *It is difficult for a human to think at more than two meta-levels at a time. Still, it is fairly easy for a human to focus on any two adjacent meta-levels.*

[3] We proposed RAAPI in 2013 by combining the best from existing repository APIs. RAAPI can be mapped to virtually any model repository. The actual version can be found at http://webappos.org/dev/raapi/.

Table 1. Some modificating RAAPI functions (actions) and their encodings.

Modificating action	Code	Encoding (action code+arguments)
`Reference createClass(String name)`	0x01	2 numbers (1 code + 1 class reference) + 1 string (name)
`createGeneralization (Reference rSubClass, Reference rSuperClass)`	0x11	3 numbers (1 code + 2 class references)
`Reference createObject(Reference rClass)`	0x02	3 numbers (1 code + 1 class reference + 1 object reference)
`includeObjectInClass(Reference rObject, Reference rClass)`	0x12	3 numbers (1 code + 1 object reference + 1 class reference)
`Reference createAttribute(Reference rClass, String name, Reference rPrimitiveType)`	0x03	3 numbers (1 code + 1 class reference + 1 type reference) + 1 string (name)
`setAttributeValue(Reference rObject, Reference rAttribute, String value)`	0x04	3 numbers (1 code + 1 object reference + 1 attribute reference) + 1 string (value)
`Reference createAssociation( Reference rSourceClass, Reference rTargetClass, String sourceRole, String targetRole, boolean isComposition)`	0x05	6 numbers (1 code + 2 class references + 1 boolean as number + 2 references for direct and inverse association ends) + 1 string (sourceRole+'/'+targetRole)
`createLink(Reference rSourceObject, Reference rTargetObject, Reference rAssociationEnd)`	0x06	4 numbers (1 code + 2 object reference + 1 association end reference); for bi-directional links only one direction is stored (the opposite link can be derived)

To comply with this conjecture, RAAPI operations are defined for two adjacent meta-levels (the model and the meta-model level). However, all repository elements (objects, classes, attributes, and associations) are identified by 64-bit references (e.g., numbers or memory pointers) regardless of their meta-level. Thus, while working with levels i and $i+1$ we can obtain some element reference and then use it when working with levels $i+1$ and $i+2$. We can even mix references from different levels (e.g., linking an object "Peter" to a class "Collie" as in Fig. 1).

Certain RAAPI operations modify the state of the repository. We call them *modificating actions*. Some of them are mentioned in Table 1, Column 1 (besides create-actions there are also corresponding delete-actions, which are not mentioned). Other operations are read-only operations for querying/traversing the repository (see Table 2, Column 1).

Now, to encode the model we use a sequence of RAAPI modificating actions. Each modificating action is assigned an integer code. Action code as well as other non-string values (numbers, references, and booleans from action arguments as

Table 2. A representative set of RAAPI operations for querying/traversing the repository

Read-only operation	Indices and keys used	Search criteria
`findClass(String name)`	s2a(name)	the first action with code 0x01 and a2s(action) equal to *name*
`findAttribute(Reference rClass, String name)`	s2a(name)	the first action with code 0x03 and the first argument equal to rClass, and a2s(action) equal to *name* or recurse into superclasses (via getIteratorForDirectSuperClasses) for derived attributes

`isDirectSubClass(Reference rSubClass, Reference rSuperClass)`	r2a(rSubClass)
	r2a(rSuperClass)
		the first action with code 0x11 and arguments rSubClass, rSuperClass

`isDerivedClass(Reference rSubClass, Reference rSuperClass)`	r2a(rSubClass)
	r2a(rSuperClass)
		the first action with code 0x11 and arguments [rSubClass, rSuperClass] or recurse into subclasses of rSuperClasses or superclasses of rSubClass

`linkExists(Reference rSourceObject, Reference rTargetObject, Reference rAssociationEnd)`	r2a(rSourceObject)
	r2a(rTargetObject)
	r2a(rAssociationEnd)
	or
	r2a(rTargetObject)
	r2a(rSourceObject)
	r2a(inverse(rAssociationEnd))
		first, we use r2a(rSourceObject), r2a(rTargetObject), and r2a(rAssociationEnd) to look for the first action with code 0x06 and arguments [rSourceObject, rTargetObject, rAssociationEnd];
		if the action not found:
		— we use r2a(rAssociationEnd) to find the first action with code 0x05 to obtain the inverse association end;
		— we use r2a(rTargetObject), r2a(rSourceObject), r2a(inverse(rAssociationEnd)) to look for the first action with code 0x06 and arguments [rTargetObject, rSourceObject, inverse(rAssociationEnd)];

`getIteratorForDirectClassObjects (Reference rClass)`	r2a(rClass)	all actions with code 0x02 or 0x12 and the first argument equal to rClass
`getIteratorForDirectSuperClasses (Reference rSubClass)`	r2a(rClass)	all actions with code 0x11 and the first argument equal to rSubClass

`getIteratorForLinkedObjects (Reference rObject, Reference rAssociationEnd)`	r2a(rObject)
	r2a(rAssociationEnd)
	or
	r2a(rObject)
	r2a(inverse(rAssociationEnd))
		all actions with code 0x06 and arguments 1 and 3 equal to [rObject, rAssociationEnd] and
		all actions with code 0x06 and arguments 2 and 3 equal to [rObject, inverse(rAssociationEnd)]
		(we use r2a(rAssociationEnd) to find the first action with code 0x05 to obtain the inverse association end)

`getIteratorForObjectsByAttributeValue (Reference rAttribute, String value)`	s2a(value)	all actions with code 0x04, the second argument equal to Attribute, and a2s(action) equal to *value*

well as the return value) are encoded as numbers stored as 64-bit IEEE doubles (see Table 1, Columns 2 and 3). The two main reasons for such encoding are:

- IEEE double is the only type for numbers supported by JavaScript in most browsers, thus, when using doubles, we can synchronize these numbers with the browser directly, without the conversion;
- the whole sequence of actions can then be stored in a single **actions** array, where each action occupies from 2 to 6 elements (thus, the *actions* array is in fact an array of variable-length mini-arrays).

Some of the modificating actions take also strings as arguments. We can assume that there is at most one string for each action (2 strings can be concatenated into one by using a delimiter, e.g., '/', see `createAssociation` in Table 1). All such strings are stored in the **strings** array in the same order as string-containing-actions (string-actions) from the *actions* array, thus, we can infer which string is associated with each actions just from the order of elements. When synchronizing, all the strings from the *strings* array are concatenated by some other delimiter and sent as one string.

An interesting fact is that our encoding stores only create-actions. When some repository element is deleted, instead of adding a new delete-action, we just delete the corresponding create-action from the *actions* arrays (and the corresponding string from the *strings* array, if any). Thus, the length of the sequence always corresponds to the size of the model.

> **Note.** Of course, this is a simplified view on the encoding. In fact, appending elements to and deleting them from an array is not trivial. Moreover, we also need some indexing to be able to iterate throughout these arrays while skipping unnecessary actions. As the next section shows, all these operations can be efficiently implemented (and the memory increases just linearly).

The server-side repository works directly with the *actions* and *strings* arrays (using a few helper arrays for efficient iterating), thus, minimizing memory consumption. The client-side repository (running in the browser) can convert the received *actions* and *strings* arrays to less efficient, but more convenient encoding using native JavaScript objects, since there is only one user at the client-side, controlling all the browser resources.

3 Implementation

The *actions* and *strings* arrays are implemented as classical resizable arrays with the amortized constant-time add and delete operations. Delete-actions are not deleted right away (which could result in shifting the arrays) – they are marked as deleted instead. When too many actions have been marked as deleted, or when there is no space for storing a new create-action, one or both arrays are re-arranged (this operation is rare compared to the cumulative number of add and delete operations). The re-arrange operation compacts the given array by shifting the elements and eliminating delete marks. Then the array length is multiplied by 0.5, 1, or 2 depending on the number of free elements in the end of the re-arranged array.

Our experiments with RAAPI show that the length of the actions array is approximately 10 times the length of the strings array. We have chosen initial lengths of 10,000 and 1,000. The arrays can grow independently up to 1,310,720,000 and 131,072,000, respectively, unless lower limits are specified[4].

3.1 Additional Data Structures

To be able to traverse the model, we introduce 3 indexing data structures (indices). The first 2 are:

- the action-to-string map **a2s** (one action can have at most one associated string);
- the inverse string-to-action multimap **s2a** (the same string can be found in multiple actions, e.g., different objects can have the same attribute value).

They allow us to implement read-only RAAPI operations that return strings (e.g., *getClassName*, *getAttributeValue*) or look up for a reference given a string (e.g., *findClass*, *findAttribute*, or *getIteratorForObjectsByAttributeValue*).

Each action is identified by a corresponding index in the actions array. Each string is identified by an index in the strings array (however, string comparison is performed not on indices, but on the actual string values from the strings array).

The third indexing structure is the reference-to-action multimap **r2a** (the same reference, e.g., object reference, can be found within multiple actions). This map allows us to traverse only actions where the given reference is used. We do not need the inverse map, since, given an index in actions array, we can instantly access the corresponding mini-array containing the action code along with all references used as action arguments[5].

Notice that all 3 indices increase memory consumption just linearly (*a2s* and *s2a* sizes are comparable to the length of the *strings* array; *r2a* size is comparable to the *actions* length). However, when re-arranging actions and strings, we have to rebuild the indices (but that still keeps the amortized time for add and delete operations constant, since re-arrange is rare operation).

[4] The first number is the maximum length of actions that does not cause integer overflow ($2^{31} - 1$), which allows us to use 4-byte integers to encode positions in the actions array. The actions array then can occupy up to 10 GB (not counting strings), which we consider quite liberal for a single model accessed by a single user via a web application.

[5] We could also embed the *a2s* map into the *actions* array by appending a string index to mini-arrays of string-actions. However, that would mix the *actions* array with the auxiliary *a2s* array. In addition, the length of the *actions* array and, hence, the amount of data synchronized with the client would increase.

Having just these 3 maps/multimaps we can implement efficiently all read-only RAAPI operations as well as certain auxiliary internal operations such as cascade delete. The following subsections provide more detail.

3.2 Querying/Iteration

Table 2 mentions a representative subset of read-only RAAPI operations and reveals which indices and keys are used to implement them. Each key is used to obtain a list of actions from some index (*r2a* or *s2a*). Then these actions are checked against the conditions mentioned in Column 3 (sometimes *a2s* is used there to check equality of strings).

As Column 2 shows, sometimes we have to look at multiple lists of actions at the same time. For some RAAPI operations (e.g., *isDirectSubClass*) we just need to get the first action that belongs to all the given lists and meets the criteria, while for other (e.g., *getIteratorForLinkedObjects*) we have to iterate through all such actions.

Good news is that all lists of actions turn out to be sorted, since each time a new action is added, it is appended to the end of the actions array (perhaps, after re-arrange), where the index of the new action is greater than the index of all previous actions. Then this action and its arguments are added to the corresponding indices *r2a*, *a2s*, and *s2a*. Thus, we can use the "merge" approach when traversing actions that must belong to multiple lists (see the listing below). To make the search within multiple lists more efficient, we implemented the *nextGreaterOrEqual* operation via binary search.

```
findFirstActionWithin(lists) {
  // initializing iterators and getting first elements of
  // the lists (iterators return INFINITY, if there are no
  // more elements)
  for (i=0; i<lists.length; i++) {
   iterators[i] = lists[i].iterator();
   values[i] = iterators[i].first(); // INFINITY, if empty
  }
  m = max(values);
  while (m<INFINITY and not all values equal m) {
    // moving forward all iterators until each of them
    // points to an element >=m or to the end of the list
    for(i=0; i<lists.length; i++)
      values[i] = iterators[i].nextGreaterOrEqual(m);
    m = max(values);
  }
  return m; // INFINITY, if at least one list ended
}
```

The indices are used not only for queries/iterations, but also in modificating actions for validating the arguments. For example, in *setAttributeValue* we have to check that the given object exists and the given attribute reference is legitime, i.e., the object belongs to a class that has that attribute defined. In addition, we have to find and delete the previous attribute value, if any.

3.3 Cascade Delete

When a delete-operation is called, we find the corresponding create-operation in the actions array and mark it as deleted (for string actions we also mark *strings*[*a2s(action)*] as deleted). However, in certain cases cascade delete is required. For example, when deleting a class, all its objects have to be deleted as well. Thus, not only the *createClass* action (0x01) has to be marked as deleted, but also all subsequent *createObject* operations (0x02) having the same class reference as the first argument. When deleting an object, all corresponding attribute values (0x04) and links (0x06) have also to be deleted (marked). All such marked actions will be cleaned up during re-arrange.

To implement cascade delete we use the same *r2a* multimap as for querying/iteration. For instance, when we mark the *createClass* operation as deleted, we obtain the class reference *rClass*. Then we obtain the list *r2a(rClass)* and iterate through it to find actions with code 0x02 (*createObject*) and reference *rClass*. For each such action we obtain the object argument *rObject* and then iterate though the *r2a(rObject)* list and mark all its elements as deleted (since *rObject* is being deleted, any action that was stored after this *createObject* and referencing the same *rObject* must be marked as deleted; this will delete 0x04, 0x06, and, perhaps, other actions having *rObject* somewhere as an argument).

3.4 Memory-Mapped Files

Although we can use standard data structures (such as Java arrays and hashmaps) for the *actions* and *strings* arrays as well as for the indices, such implementation quickly leads to high dynamic memory consumption (and even to out-of-memory exception, if more than 110 middle-sized repositories are open, see below). This is undesirable for the web server. Our approach is to rely on memory-mapped files, a mechanism, which is available in most operating systems. The OS automatically swaps memory pages, while the programmer can access the data via a single pointer as if the data were always loaded into memory. With memory-mapped files, server memory is not limited to the size of the physical RAM, and the OS does all the low-level job automatically and efficiently (for instance, files are loaded into memory in lazy manner, thus attaching a file as a pointer is fast). The shortcoming is that memory-mapped files, in essence, are arrays. While the actions array can be mapped directly to a file, other data structures (indices and strings) have to be mapped to arrays manually.

To be able to store strings in a memory-mapped file, we use 2 arrays: **chars** and **$\mathbf{strings_2}$**. The first one is for appending characters of each new string (we use UTF-8 character encoding); the second one stores the start index in the *chars* array and the string length (in bytes). The re-arrange functions works only on the relative short $strings_2$ array, thus, characters are not moved[6].

[6] The *chars* array is much longer than $strings_2$. Thus, to save time during re-arrange, the *chars* array can be left "as is", without removing characters of deleted strings (still, it can be compacted occasionally, e.g., during save).

The *r2a*, *a2s*, and *s2a* indices are implemented as arrays of keys and values. The lengths of these arrays are prime numbers that depend on the lengths of the *actions* and $strings_2$ arrays. Prime lengths allow us to use these arrays as hash tables with open addressing and double hashing [7] [12]. Since *r2a* and *s2a* are multimaps, we modify traditional hashing approach: for multi-valued keys we store a negative number $-(k+1)$ in a hash table, where k is the number of values already stored for this key (including the collisions). Thus, to append a new value for the given key, we first skip $(k+1)$ elements and try to append the value as usual. Our experiments show that the number of collisions for such multimaps (when working on a repository containing data from a real use case) is 2.28 in average.

While deleting elements from a hash table may be non-trivial, our approach is simple: we just mark elements as deleted (when the corresponding actions are marked as deleted). During re-arrange, hash tables are rebuilt from scratch. However, this approach introduces a new issue: when traversing the values of a multimap, we can encounter such marked-as-deleted elements. If we need to iterate through all elements, we can just ignore these marked elements. However, the function *nextGreaterOrEqual* mentioned above won't work any more, since the sorted list of values now can contain deleted (marked) values, and the binary search algorithm won't work as expected. Generally speaking, the binary search has to be replaced with linear search[8]. However, since the number of marked elements is small (otherwise, the array is re-arranged), we introduce the following modification of the binary search operation: when we encounter a marked-as-deleted element that should become a new middle element, we look for the next non-marked element linearly. Then the search continues as ordinary binary search. This modification proved to be very fast in practice (it boosted model transformations by 60.56% compared to fully linear implementation of *nextGreaterOrEqual*).

4 Feasibility

In this section we provide details on CPU and memory benchmarks. We also give some notes on synchronization overhead.

4.1 CPU Benchmark Tests

Table 3 provides averaged CPU benchmark data for the proposed repository AR (acronym for "Actions Repository") in comparison with Ecore and JR [18,20].

[7] The first hash is modulo p, the second is modulo $(p-2)+1$, which is always co-prime with p. For strings we use Java built-in *hashCode* function. However, since it returns 0 on empty strings, and since the second hash calculates to 1, all empty strings would be stored in the beginning of the hash table, thus, drastically increasing the number of collisions (up to 2.85x in our experiments). We avoid such inefficient hash values by appending a constant dummy text to every string before calculating its hash.

[8] Grover's algorithm on a real quantum computer could take sub-linear time, but the proposed repository is intended for classical computers.

Table 3. CPU benchmark (all values are in milliseconds per repository)

	ECore	**JR**	**AR (Java hash maps)**	**AR (hash tables)**	**AR (memory-mapped files)**
Repository time	1016	857	423	618	760
Overhead time	20,015*	436	106	93	85

Processor Intel i7-2600, 3.40 GHz, repository running within a single thread, 64-bit Java Virtual Machine 1.8.0 on Windows, no heavyweight parallel processes running.

Profiler: Java VisualVM 1.8.0 in CPU profiling mode.

Repository time accuracy w.r.t. the mean value is 10%, overhead time accuracy is 25%.

*Due to ECore internal design, implementation of some RAAPI operations required significant overhead in order to avoid ECore exceptions.

AR and ECore are implemented in Java, while JR—in plain C (until now, JR proved to be the fastest repository we ever used in our model-based tools).

In our tests we were interested in 3 variations of AR: using Java standard data structures (Java arrays, HashMaps, and ArrayLists), using hash tables implemented manually via in-memory arrays, and using hash tables stored in a memory-mapped file. In all cases we used a transformation borrowed from the ontology editor OWLGrEd (http://owlgred.lumii.lv). The transformation we chose performs a set of actions (such as creating a dialog window from a model, storing the input in the repository, and refreshing the diagram from the updated model) that represent a real usage step of a graphical model-based tool. We measured not only CPU clock for each of the repositories, but also the overhead added by wrappers, which map universal RAAPI to native repository APIs (operations not provided by native APIs were implemented in wrappers). We can infer from Table 3 that AR outperforms both ECore and JR. Logically, memory-mapped files are a bit slower than direct in-memory hash tables. Java built-in data structures show the best CPU benchmark rates (but not the best memory rates, as is shown below).

4.2 Memory Benchmark Tests

Table 4 provides averaged memory benchmark for the repository, on which the transformation mentioned above was executed.

We measured not only memory consumption, but also repository load time. For memory-mapped files we split our tests into 2 groups: tests from the first group were executed, when there were no memory-mapped files on disk (thus, they had to be created by the OS and filled with data by AR); tests from the second group just opened existing memory-mapped files. As Table 4 shows, AR with memory mapped files was the only repository that could handle 10,000 models at the same time with significant room for scaling (and the OS reserved just 54 MiB of RAM for all of them, if we do not count the files on disk amounting to 122 GiB in total). We have to admit that our tests did not include heavyweight

Table 4. Repository memory usage (MiB/repository) and open time (ms/repository)

Number of repositories		JR	ECore	AR (Java hash maps)	AR (hash tables)	AR (creating memory-mapped files); 12.2MiB on disk per repository	AR (opening memory-mapped files); 12.2MiB on disk per repository
100	**memory**	33.41$^{\bowtie}$	12.77	13.98	11.25	1.67	0.013
	time	1047$^{\bowtie}$	849	153	63	61 (SSD) or 69 (HDD)	26 (SSD) or 9 (HDD)
1,000	**memory**	n/a	9.45$^{\diamond}$ or 9.57$^{\diamond\diamond}$	13.74* or 13.55**	10.72$^{\square}$	0.071	0.013
	time		4763$^{\diamond}$ or 5414$^{\diamond\diamond}$	189* or 714**	83$^{\square}$	68 (SSD) or 148 (HDD); 420 (LAN)$^{\natural}$	27 (SSD) or 33 (HDD); 280 (LAN)$^{\natural}$
10,000	**memory**	n/a	n/a	n/a	n/a	0.012	0.005
	time					226 (HDD)	41 (HDD)

Profiler: Java VisualVM 1.8.0 in CPU and memory profiling modes.

$^{\bowtie}$ One JR instance; it is impossible to open multiple JR instances in the same process.
$^{\diamond}$ 173 repositories before out of memory; automatic garbage collection
$^{\diamond\diamond}$ 180 repositories before out of memory; forced garbage collection
* ≈111 repositories before freeze; automatic garbage collection
** 101 repository before out of memory; forced garbage collection
$^{\square}$ ≈142 repositories before out of memory; automatic garbage collection
$^{\natural}$ Windows share (samba) over a 100 Mbit/s local area network

parallel processes or intensive usage of memory by multiple users with inevitable competition for processor cache (these tests are subject to additional research). Nevertheless, current results look promising.

4.3 Synchronization

CPU and memory efficiency is not enough for a repository with web-based usage in mind. We have to be able to synchronize the repository efficiently between the client and the server. Our solution relies on using web sockets, a standardized protocol with low overhead (when set up properly, web sockets can hold 1,000,000 connections, and even more). Since web sockets can be used to transmit both binary and string data, the actions and strings can be synchronized efficiently. The client and the server can modify the repository independently, each on its side. We use the following trick to avoid collisions in references: when new repository elements (objects, classes, associations...) are created, the server assigns even references for them, but the client assigns odd. Modifications are exchanged asynchronously (both at the server and at the client side), thus, a separate thread is busy with that, while the original thread running a model transformation continues without any delay (delay can only be caused by network buffers overflow). To optimize the synchronization process, we collect several modifications within a small time interval and then send them in bulk. Modifications are sent using the encoding of the *actions* and *strings* arrays with an exception that modifications can contain also delete-actions. When received, modifications are re-executed on the receiving side as if they occurred right there.

5 Discussion

Currently, AR iterators over repository elements are not thread-safe internally. We deal with this issue by synchronizing public RAAPI calls and copying the required elements each time an iterator is returned via RAAPI. In the future, to boost iterators, we could switch to the copy-on-write pattern (where copying is done only if a parallel modification is performed).

Since all elements (classes, associations, etc.) in AR are identified by 64-bit references, we can create classes and objects at different meta-levels and even mix them. Thus, AR can be used for storing models corresponding to virtually any meta-modelling standard (e.g., MOF, EMOF, or SMOF) or ontology language (e.g., OWL or OWL2) [2,3,15,16]. This can lead to interesting use cases. For instance, we can create meta-meta-level classes corresponding to the OWL2 standard (*OWL:Class*, *OWL:Property*, etc.). Then we can create ordinary metamodel classes and call *includeObjectInClass* to make them instances of the meta-meta-level classes (e.g., class *Person* would become an instance of *OWL:Class*). All these operations are legitime and are just added to the actions array. Then, by using AR indices, we can infer which classes are instances of OWL2 meta-metamodel, and forward them to a semantic reasoner.

AR can also be used in a NoSQL-manner, where the metamodel is not defined in advance. This can be implemented in 2 ways:

- by skipping metamodel checks (i.e., not validating action arguments);
- by introducing a wrapper. When some action requiring a metamodel element is performed, the wrapper creates a missing metamodel element on-the-fly. This, however, requires advanced techniques for guessing metamodel elements (e.g., guessing types of attributes or inheritance relations) and, perhaps, modifying them dynamically, if eventually we find that the initial guess was incorrect.

Our tests showed that AR is more efficient than JR. The JR authors showed that their repository outperforms popular *OpenLink Virtuoso*. Pacaci et al. showed that *Virtuoso* outperforms other graph databases, and Hellerstein et al. showed that graph databases outperform relational ones [4,19]. While these facts may seem to be in favor to the proposed repository, we have to admit that the performance depends on a particular usage scenario. For instance, Pacaci et al. showed that traditional relational *Postgres* database outperformed *Virtuoso* in several specific tests [19].

It is hard to compare AR to linked data and their query mechanisms (like *Linked Data Fragments*, linkeddatafragments.org), since they are optimized for single-query usage (where each query can be quite complex), while AR is designed to serve multiple, but simple queries performed by model transformations.

6 Conclusion

We presented a model repository that outperforms existing repositories regarding both CPU and memory. The main idea was to use an efficient encoding of the model by storing a list of actions (and corresponding strings) that create the content of the repository (which resembles the Kolmogorov complexity concept). We added just 3 indexing arrays for implementing RAAPI query and iteration operations. The proposed encoding, combined with memory-mapped files, can hold 10,000 repositories (and even more) on a single server. That resembles the C10K problem (10,000 concurrent connections; that is considered a reasonable target for web-based applications[9]) [10]. However, stress tests concerning CPU cache and context switches still have to be performed.

The proposed repository encoding is used "as is", when synchronizing the repository via the network (the encoding even uses the IEEE double as the only JavaScript-compatible type for numbers). Since we use asynchronous web sockets, synchronization overhead is negligible (unless the network becomes a bottleneck).

The repository implements universal RAAPI, where the developer thinks at two adjacent meta-levels (the model and the meta-model level), but can use any number of meta-level and even mix them.

We hope the repository will find wide adoption, thus, we release it under an open-source license[10]. The repository is written in Java, but a dynamic-link library for accessing it from native code is available (32-bit and 64-bit versions for Windows, Linux, and MacOS platforms).

We are working on developing a model-based infrastructure for web applications (webAppOS), where the proposed repository will be a central component implementing memory abstraction. A webAppOS-based version of our graphical ontology editor OWLGrEd is coming soon. OWLGrEd diagrams will be stored using AR, making the proposed repository a part of the new OWLGrEd file format for both desktop and web-based versions of OWLGrEd.

Acknowledgments. The work has been supported by European Regional Development Fund within the project #1.1.1.2/16/I/001, application #1.1.1.2/VI-AA/1/16/214 "Model-Based Web Application Infrastructure with Cloud Technology Support".

[9] For more connections or during peek loads, one can borrow virtual cloud servers, e.g., from Amazon Elastic Cloud.

[10] The repository can be downloaded at http://webappos.org/dev/ar.

References

1. Eclipse Modeling Framework (EMF, Eclipse Modeling subproject). http://www.eclipse.org/emf
2. OWL 2 Web Ontology Language document overview (second edition). http://www.w3.org/TR/owl2-overview/
3. OWL Web Ontology Language reference. http://www.w3.org/TR/owl-ref/
4. Hellerstein, J.M., et al.: Ground: a data context service. In: Proceedings of CIDR (2017)
5. Ambler, S.: Mapping objects to relational databases: O/R mapping in detail. http://www.agiledata.org/essays/mappingObjects.html
6. Anuja, K.: Object Relational Mapping. Ph.D. thesis, Cochin University of Science and Technology (2007)
7. Barzdins, J., Kalnins, A., Rencis, E., Rikacovs, S.: Model transformation languages and their implementation by bootstrapping method. In: Avron, A., Dershowitz, N., Rabinovich, A. (eds.) Pillars of Computer Science. LNCS, vol. 4800, pp. 130–145. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-78127-1_8
8. Jouault, F., Kurtev, I.: Transforming models with ATL. In: Bruel, J.-M. (ed.) MODELS 2005. LNCS, vol. 3844, pp. 128–138. Springer, Heidelberg (2006). https://doi.org/10.1007/11663430_14
9. Kalnins, A., Barzdins, J., Celms, E.: Model transformation language MOLA. In: Aßmann, U., Aksit, M., Rensink, A. (eds.) MDAFA 2003-2004. LNCS, vol. 3599, pp. 62–76. Springer, Heidelberg (2005). https://doi.org/10.1007/11538097_5
10. Kegel, D.: The C10K problem. http://www.kegel.com/c10k.html
11. Kemp, S.: Digital in 2018: World's internet users pass the 4 billion mark. wearesocial.com blog. https://wearesocial.com/blog/2018/01/global-digital-report-2018
12. Knuth, D.E.: The Art of Computer Programming, Sorting and Searching, 2nd edn., vol. 3. Addison Wesley Longman Publishing Co., Inc., Redwood City (1998)
13. Kolovos, D., Rose, L., Paige, R.: The Epsilon Book. http://www.eclipse.org/epsilon/doc/book/
14. Kozlovics, S.: The orchestra of multiple model repositories. In: van Emde Boas, P., Groen, F.C.A., Italiano, G.F., Nawrocki, J., Sack, H. (eds.) SOFSEM 2013. LNCS, vol. 7741, pp. 503–514. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35843-2_43
15. Object Management Group: OMG Meta Object Facility (MOF) Core Specification Version 2.4.1 (2011)
16. Object Management Group: MOF Support For Semantic Structures (SMOF) (2012). http://www.omg.org/spec/SMOF/
17. Object Management Group: Meta Object Facility (MOF) 2.0 Query/View/Transformation Specification, Version 1.3. formal/16-06-03 (2016)
18. Opmanis, M., Čerāns, K.: Multilevel data repository for ontological and metamodeling. In: Databases and Information Systems VI - Selected Papers from the Ninth International Baltic Conference, DB&IS 2010 (2011)
19. Pacaci, A., Zhou, A., Lin, J., Özsu, M.T.: Do we need specialized graph databases?: benchmarking real-time social networking applications. In: Proceedings of the Fifth International Workshop on Graph Data-management Experiences & Systems, GRADES 2017, pp. 12:1–12:7. ACM, New York (2017)
20. Steinberg, D., Budinsky, F., Paternostro, M., Merks, E.: EMF: Eclipse Modeling Framework, 2nd edn. Addison-Wesley, Upper Saddle River (2008)
21. Varró, D., Balogh, A.: The model transformation language of the VIATRA2 framework. Sci. Comput. Program. **68**(3), 187–207 (2007)

Big Data Analysis and Processing

Scalable Hadoop-Based Infrastructure for Big Data Analytics

Irina Astrova[1], Arne Koschel[2], Felix Heine[2], and Ahto Kalja[1](✉)

[1] Department of Software Science, School of IT, Tallinn University of Technology,
Akadeemia tee 21, 12618 Tallinn, Estonia
{irina,ahto}@cs.ioc.ee

[2] Faculty IV, Department of Computer Science,
Hannover University of Applied Sciences and Arts,
Ricklinger Stadtweg 120, 30459 Hannover, Germany
akoschel@acm.org, felix.heine@hs-hannover.de

Abstract. Cloud architectures are being used increasingly to support Big Data analytics by organizations that make ad hoc or routine use of the cloud in lieu of acquiring their own infrastructure. On the other hand, Hadoop has become the de-facto standard for storing and processing Big Data. It is hard to overstate how many advantages come with moving Hadoop into the cloud. The most important is scalability, meaning that the underlying infrastructure can be expanded or contracted according to the actual demand on resources. This paper presents a scalable Hadoop-based infrastructure for Big Data analytics, one that gets automatically adjusted if more computing power or storage capacity is needed. Adjustments are transparent to the users – the users seem to have nearly unlimited computation and storage resources.

Keywords: Big Data · Cloud computing · Hadoop

1 Introduction

Big Data are big in two different senses. They are big in the quantity and variety of data that are available to be stored and processed. They are also big in the scale of analysis (or analytics) that can be applied to those data. Both kinds of "big" depend on the existence of supportive infrastructure. Such an infrastructure is increasingly being provided by the cloud like OpenStack or Amazon EC2.

The DC4C (Data Cloud for Cities) project was initiated at Hannover University of Applied Sciences and Arts. The primary goal of the DC4C project was to create a cloud-based infrastructure for Big Data analytics. An initial step toward this goal was to build a high-level architecture for such an infrastructure. A next step forward was to implement a low-level architecture based on Hadoop clusters running in the cloud.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 233–242, 2018.
https://doi.org/10.1007/978-3-319-97571-9_19

2 Motivation

A Hadoop [1] cluster or more specifically HDFS (Hadoop Distributed File System) cluster consists of a *NameNode* to store the HDFS metadata and an arbitrary number of *DataNodes* that store data. Data are split into blocks and then replicated at multiple *DataNodes* to ensure high availability and performance. For data processing, Hadoop employs the MapReduce programming model, where a master node coordinates a number of *WorkerNodes* that do the actual processing.

The virtualization of a Hadoop cluster has many benefits. If a Hadoop cluster runs in the cloud, the physical hardware can be shared with other components like a web server or a relational database server to utilize the full performance of the hardware. Another advantage of a virtualized Hadoop cluster is that the setup of the cluster can be simplified by creating templates of the virtual machines, which can be easily started and stopped, thus avoiding the need of a time-intensive installation routine. This comes in conjunction with yet another benefit – a virtualized Hadoop cluster can be rapidly expanded or contracted depending on the current demand on resources.

Furthermore, all general benefits of the cloud can be applied to a virtualized Hadoop cluster like the high-availability of virtual machines or the ability to clone images for using them as a backup or node instances. On the other hand side, Hadoop is designed to offer things like reliability and high-availability by default but other things like the full utilization of the hardware by sharing them with other components bring a real benefit. Finally, not only the hardware could be shared with different components but also with another Hadoop distribution. The benefits are quite appealing but it also brings some challenges, which need to be addressed.

3 Challenges

The cloud computing technology is based on the concept of virtualization. However, the virtualization of a Hadoop cluster is a challenging task. Unlike most common server applications, Hadoop has some special requirements for a cloud architecture. In particular, Hadoop requires for topology information about the utilized infrastructure. Hadoop then uses this information to manage replications in HDFS. If the infrastructure consists of more than one cluster, Hadoop ensures that at least one replication is stored in a different hardware cluster than the other replications to allow for data access even when the whole cluster is unavailable. Moreover, Hadoop tries to perform computing tasks near the required data to avoid network transfers, which are often slower than local data access.

A cloud architecture abstracts the physical hardware to hide the infrastructure details from the hosted instances. Furthermore, shared storage pools are often used to store instances instead of having a dedicated storage in every computing node. Shared storage pools and missing topology information of the Hadoop instances might lead to multiple HDFS replications onto the same physical storage pool. Also Hadoop's paradigm to avoid network traffic by allocating computing tasks near the data storage would be broken, since shared storage pools are often connected via a network. As a consequence, the performance of cluster would probably be massively decreased due to unnecessary replications and increased network traffic.

4 High-Level Architecture

Figure 1 gives an overview of the high-level architecture. This architecture is multilayered – different layers allocate the different responsibilities of Big Data software. An upper layer uses a lower layer as a service.

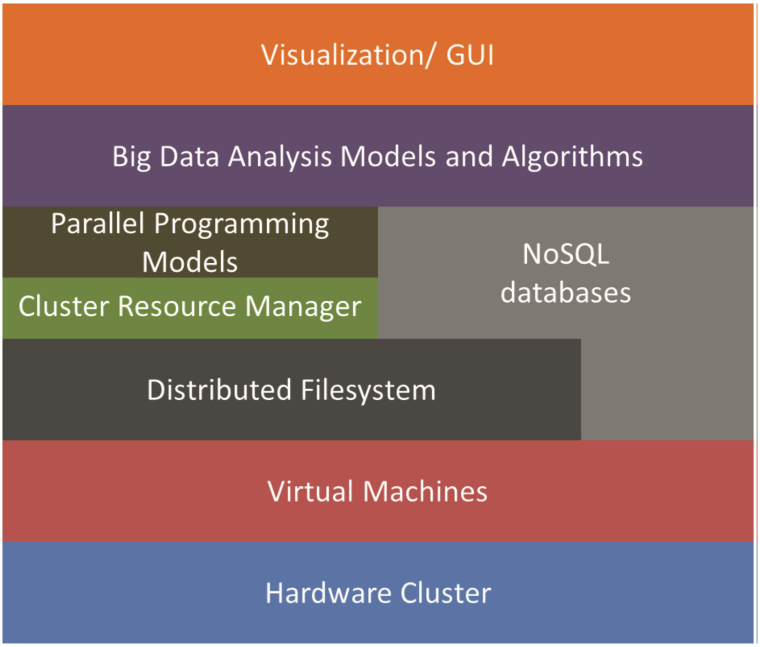


Fig. 1. High-level architecture

The architecture comprises the following layers:

- **Hardware (or physical) clusters:** This is the bottom layer. It is composed of all physical machines that are wired together either by the Internet or by a direct network connection. A physical cluster is abstracted by means of virtualization.
- **Virtual machines:** This layer is composed of all virtual machines.
- **Distributed file system:** This layer is composed of a distributed file system used for storing Big Data (typically in the range of gigabytes to terabytes) that are distributed across the virtual machines.
- **NoSQL databases:** This layer is composed of NoSQL databases like HBase. Being installed on the top of the distributed file system, NoSQL databases make it easy to create, retrieve, update and delete data using an SQL-like language. In addition, some NoSQL databases can be installed directly on the virtual machines.

- **Cluster resource manager:** This layer is responsible for managing both the physical and virtual clusters using an API (Application Programming Interface).
- **Big Data analytics models and algorithms:** This is the application layer that is responsible for Big Data analytics.
- **Parallel programming models:** The applications implement algorithms that are parallelized using programming models like MapReduce [2] and Giraph Pregel [3].
- **Visualization and GUI:** This is the top layer of the architecture. A graphical user interface (GUI) provides simple and easy access to Big Data. Furthermore, visualization improves the understanding of the results of Big Data analytics.

5 Low-Level Architecture

Figure 2 gives an overview of the low-level architecture.

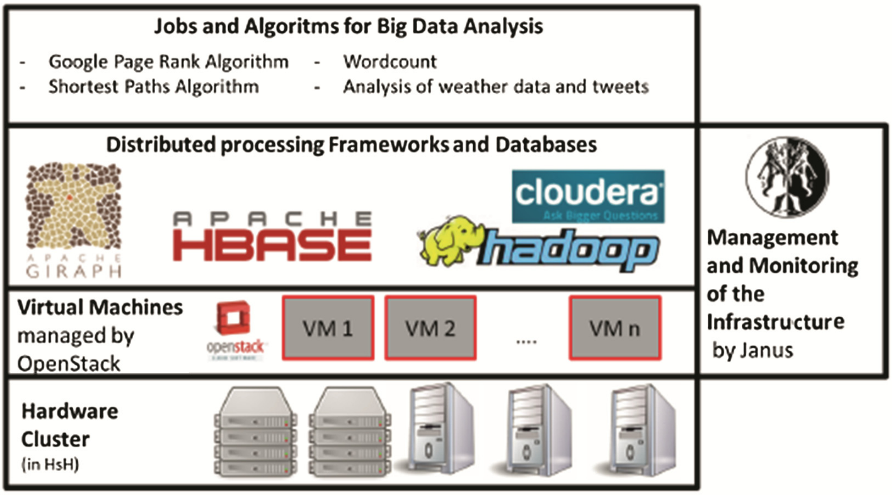

Jobs and Algoritms for Big Data Analysis
- Google Page Rank Algorithm
- Shortest Paths Algorithm
- Wordcount
- Analysis of weather data and tweets
Distributed processing Frameworks and Databases
APACHE GIRAPH
APACHE HBASE
cloudera
hadoop
Virtual Machines managed by OpenStack
openstack
VM 1
VM 2
....
VM n
Hardware Cluster (in HsH)
Management and Monitoring of the Infrastructure by Janus

Fig. 2. Low-level architecture

The architecture comprises the following layers:

- **Hardware (or physical) clusters:** This layer is composed of five servers (Quad core Xeon, 16 GB RAM, 3 TB HDD) and three servers (Quad core Xeon, 128 GB RAM, 3 TB HDD).
- **Virtual machines:** On top of the hardware clusters, we installed virtual machines to form a hardware abstraction layer. A virtual machine acts like a physical computer except that software running on the virtual machine is separated from the underlying hardware resources. This layer is managed by OpenStack, which eases the management of virtual machines by a standardized API.

- **Distributed processing frameworks and databases:** This layer is composed of Hadoop, HDFS, HBase, MapReduce, Giraph Pregel and Cloudera. Cloudera is a distribution, which delivers many Apache products, including Hadoop and HBase.
- **Jobs and algorithms for Big Data analytics:** On this layer, we implemented many algorithms like Google's PageRank, Shortest Paths and Word Count.
- **Management and monitoring of the infrastructure:** On this layer, we implemented Janus, which monitors and tracks the virtual machines to react to storage or computing capacity bottlenecks. Janus provides an API to automate the launching, management and resizing of Hadoop clusters.

6 Implementation

It could be beneficial to co-locate the allocations of a job on the same rack (affinity constraints) to reduce network costs, but spread the allocations across machines (anti-affinity constraints) to minimize resource interference. If multiple Hadoop instances (viz., *DataNodes*) are placed on the same machine, replicated data will be-come unavailable if that machine fails.

Janus has to take care of the anti-affinity of Hadoop instances as well as to monitor both clusters (the physical cluster as well as the virtual cluster, which runs inside the physical cluster). The physical cluster is controlled by OpenStack whereas the virtual cluster is controlled by Cloudera Manager. To perform these tasks, Janus has to work with both managers. One virtual machine host should never run more than one instance from the same Hadoop cluster. This would endanger the redundancy of HDFS and might decrease the general performance with unnecessary replications. Instances forming a Hadoop cluster are anti-affine to all other instances of the same cluster but not anti-affine to instances of other Hadoop clusters. This means that one host system could generally run more than one Hadoop instance, but they cannot be from the same Hadoop cluster.

Figure 3 shows an overview of Janus architecture.

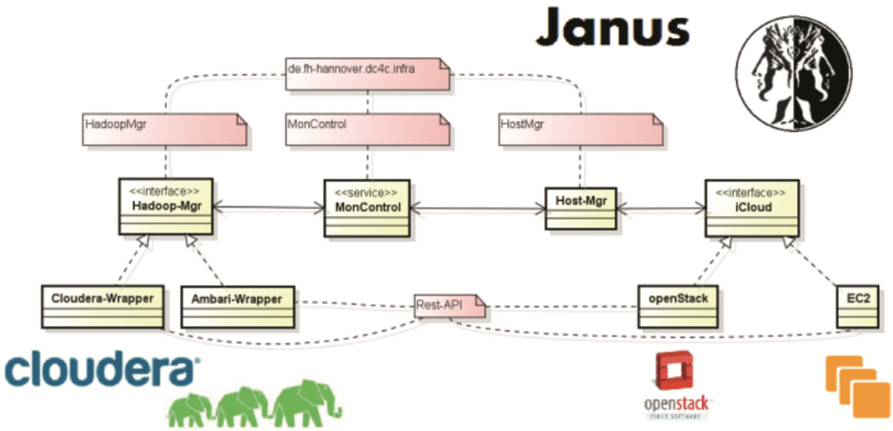


Fig. 3. Architecture of Janus [6]

The architecture mainly consists of the following components:

- **Host manager:** This component is represented by a `Host-Mgr` class with the following methods:
  - `getHadoopClusters()` returns a list of all currently used Hadoop clusters.
  - `launchHadoopCluster()` creates a new Hadoop cluster with a given number of nodes.
  - `extendHadoopCluster()` adds a given number of new nodes to the Hadoop cluster.
  - `shrinkHadoopCluster()` removes a given number of nodes.
- **Cloud manager:** This component is represented by an `iCloud` interface with the following methods:
  - `getHosts()` returns a list of all physical machines.
  - `getInstances()` returns a list of all virtual machines.
  - `createInstance()` creates a new virtual machine.
  - `deleteInstance()` deletes the virtual machine.
- **Hadoop manager** (which connects the two other managers): This component is represented by a `Hadoop-Mgr` interface with the following methods:
  - `getServices()` returns a list of all services.
  - `getServiceDetails()` returns information on a given service.
  - `addNode()` creates a new node.
  - `deleteNode()` deletes the node. This includes the decommissioning of all the services running on that node and only then the deletion of the node itself.
  - `getNodeStatus()` returns the status of a given node.

Janus is a link between the cloud manager of OpenStack and the Cloudera Manager. It connects both sides by utilizing classes, which implement two abstract interfaces: `iCloud` and `Hadoop-Mgr`. This is done to ensure that the core logic of Janus will not be altered even if cloud providers are added or removed. On the cloud management side, these are the classes of OpenStack or Amazon EC2, which implement the `iCloud` interface. On the Hadoop cluster management side, these could be the Hortonworks Ambari Manager Wrapper or the Cloudera Manager Wrapper classes, which implement the `Hadoop-Mgr` interface.

In Hadoop, there are several different roles, including *DataNode* and *WorkerNode*. A role is an instance of the service that is bound and executed on a host. As the names suggest, the *DataNode*'s role is to store incoming data, the *WorkerNode*'s role is to process the data. In most cases, these roles are combined into one node. To react dynamically to the fluctuation of the amount of incoming data, which a Hadoop cluster receives, nodes have to be created or deleted on demand. For example, when HDFS runs out of space, Janus executes the appropriate methods of the `iCloud` interface to initiate the creation of a new node with the two roles. Janus makes a synchronous call and waits for the result of the method call. If the new node is created successfully, it will execute the appropriate methods of the `Hadoop-Mgr` interface to initiate the addition of the new node to the appropriate cluster of the Cloudera Manager.

To start the roles, we used a predefined template. This template was created manually on the Cloudera Cluster setup so it can be used to create new nodes. The template was named `DC4C-Default-HDFS-MR` and included two roles: `mapreduce`, which is the *WorkerNode*, and `HDFS`, which is the *DataNode*. The template has to be called by its name and is passed to the `addNode` method of the `ClouderaManagerResource` class as a parameter. The call of the method applies the template to the newly created host and starts the roles.

To avoid the need of an additional database, Janus enforces a strict naming scheme for the Hadoop instances, which allows for the mapping only by the hostnames. Upon start up, Janus loads information about all hosts in its managed clouds via API calls to the OpenStack masters and maps the currently running instances to the hosts. Hostnames, which do not fit the naming schema, are ignored so that additional instances for other purposes can be managed manually.

7 Application Scenarios

We identified two major application scenarios for the implemented architecture. One was about the storage capacity offered to the users. If the storage capacity becomes scarce, the infrastructure will automatically increase the size of the HDFS. If the physical storage limit of the hardware gets reached, the infrastructure will automatically contact another cloud provider. This could be a private cloud provider like another university or partner organization or a public cloud provider like Amazon. The users get the possibility to define prioritization of external clouds to minimize the expenses, which arise when commercial public clouds are used. Whenever a Hadoop cluster needs to be extended, Janus searches for a new suitable host system in all managed the OpenStack clouds. Thereby currently used clouds are preferred, so that a Hadoop cluster will only be extended into a new cloud as a last resort. Within each managed cloud, Janus searches for hosts, which are currently not used by the Hadoop cluster that should be expanded. If more than one host system could run a new instance, the host with the lowest count of running instances is selected. In either case, such expansion should be considered as a temporal and rapid solution to prevent data loss, which would occur if the cloud could not store any further data. In the long term, it would be necessary to buy additional hardware to offer more storage capacity within the cloud to release expensive public cloud instances.

In the DC4C project, we had two separate OpenStack installations, each having one OpenStack master and several OpenStack computing nodes. The first cloud consisted of five servers with a quad-core processor and 16 GB RAM. The second cloud consisted of three servers with a quad-core processor and 128 GB RAM. All the servers were utilizing a local RAID-0 array as their data storage to ensure highest storage performance. Redundancy was achieved by the internal replication mechanisms of HDFS. To simulate the scaling mechanism into a public cloud, we configured Janus to treat the second cloud as the public cloud. Janus broke with the anti-affinity of instances in a Hadoop cluster in the simulated public cloud and launched new instances wherever resources were available.

Another application scenario concerned the computation power of the cloud. As more and more different users would use the cloud for their Big Data analytics, a single job could get really slow if the virtual computing nodes reach their limits. The solution for this scenario is to start further virtual computing nodes in the cloud to take over additional analysis jobs. If the physical limits of the hardware also get reached, additional computation power will be obtained from an external cloud. An important point, which has to be taken into consideration when expanding into a public cloud, is the storage location of sensitive data. The users may want not to offer those data to a public cloud provider just because the infrastructure is running out of storage. In this case, the users are given the possibility to mark their data as sensitive so that the infrastructure can avoid the exposure of those data. To realize this scenario, the cloud solution has to move other non-sensitive data to the public cloud to free storage for the sensitive data.

Based on the application scenarios, we created two rules to react to storage or computing capacity bottlenecks. Both rules are checked in a cyclic interval. If a rule gets violated, the defined action will be started and no further checks of any other rule are done until the violation is fixed. One rule is `HdfsCapacityRule`. This rule is used to monitor the free disk space in HDFS; it creates a new Hadoop node if a given threshold is violated for a given timeframe. Another rule is `MapRedSlotsRule`. It is triggered when a given percentage of the available MapReduce slots have already been in use. When this rule is violated for a given timeframe, a new Hadoop node is created too.

8 Related Work

Cloud providers have started to offer prepackaged services that use Hadoop under the hood, but do most of the cluster management work themselves. The users simply point the services to data and provide the services with jobs to run, and the services handle the rest, delivering results back to the users. The users still pay for the resources used, as well as the use of the services, but save on all of the management work [5].

Examples of prepackaged services include:

- **Elastic MapReduce:** This is Amazon Web Services' solution for managing Hadoop clusters through an API. Data are usually stored in Amazon S3 or Amazon DynamoDB. The normal mode of operation for Elastic MapReduce is to define the parameters for a Hadoop cluster like its size, location, Hadoop version and variety of services, point to where data should be read from and written to, and define steps to run jobs. Elastic MapReduce launches a Hadoop cluster, performs the steps to generate the output data and then tears the cluster down. However, the users can leave the cluster running for further use and even resize it for greater capacity.
- **Google Cloud Dataproc:** It is similar to Elastic MapReduce, but runs within Google Cloud Platform. Data are usually stored in Google Cloud Storage.
- **HDInsight:** This is Microsoft Azure's solution, which is built on top of Hortonworks Data Platform. HDInsight works with Azure Blob Storage and Azure Data Lake Store for reading and writing data used in jobs. Ambari is included as well for cluster management through its API.

Despite their advantages like ready availability and ease of use, prepackaged services only work on the cloud providers offering them. Some organizations are worried about being "locked in" to a single cloud provider, unable to take advantage of competition between the providers. Moreover, it may not be possible to satisfy data security or tracking requirements with the services due to a lack of direct control over the resources [5].

In addition to prepackaged services, cloud providers offer Hadoop-as-a-Service (HaaS) for Big Data analytics [6]. However, HaaS offerings share the same disadvantages as prepackaged services in terms of moving further away from the open source world and jeopardizing interoperability. Moreover, since unlike to prepackaged services they are not explicitly based on Hadoop, there is a separate learning curve for them, and the effort could be wasted if they are ever discarded in favor of an application that works on Hadoop or on a different cloud provider [5].

9 Conclusion

Big Data analytics requires not just algorithms and data, but also physical platforms where the data are stored and processed. This class of infrastructure is now available through the cloud.

The DC4C project was aimed at developing a cloud-based infrastructure for Big Data analytics, which gets automatically adjusted if more computing power or storage capacity is needed. One of the main challenges in the development of such an infrastructure was the integration of Big Data software framework like Hadoop into a cloud architecture as both are designed for contrary purposes. Moreover, a Big Data software framework is usually complex and its usage requires a lot of practice, knowledge and experience.

The initial result of the DC4C project was a high-level architecture for the infrastructure. A major result was the implementation of a low-level architecture based on Hadoop clusters running in the OpenStack cloud. That architecture enabled to make Hadoop clusters virtualized and scalable on demand. Furthermore, the architecture was used to evaluate the performance of different persistence layers and computational models for processing data. More specifically, the persistence layers were evaluated by comparing the storage of data onto HDFS and HBase, whereas computational models were compared by executing the PageRank algorithm with MapReduce and Giraph Pregel [4]. Finally, the architecture was recognized as being worthy of application in the area of Estonian e-Government, which also needs to deal with Big Data analytics [7, 8].

10 Future Work

In the current version of Janus, rules monitor only a single property of Hadoop cluster and check if that property violates a certain threshold. These rules were primarily used to prove that a Hadoop cluster can be automatically adjusted when the CPU usage per node increases or the HDFS capacity gets too low. A future work could be to extend the existing rule engine to support rules, which monitor multiple properties. Another enhancement could be to monitor complex events occurred in a Hadoop cluster, e.g., when every Friday night a weekly computation is started and from week to week the performance gets lower. For this purpose, historical measurements have to be stored and evaluated.

Acknowledgement. Irina Astrova's and Ahto Kalja's work was supported by the Estonian Ministry of Education and Research institutional research grant IUT33-13.

References

1. White, T.: Hadoop: The Definitive Guide, 3rd edn. O'Reilly Media, Sebastopol (2012)
2. Shook, A.: MapReduce Design Patterns. O'Reilly Media, Sebastopol (2013)
3. Malewicz, G., Matthew, A., Bik, A., Dehnert, J., Horn, I., Leiser, N., Czajkowski, G.: Pregel: a system for large-scale graph processing. In: Proceedings of the 2010 ACM SIGMOD International Conference on Management of Data, New York, USA (2010)
4. Koschel, A., Heine, F., Astrova, I., Korte, F., Rossow, T., Stipkovic, S.: Efficiency experiments on Hadoop and Giraph with PageRank. In: Proceedings of 24th Euromicro International Conference on Parallel, Distributed, and Network-Based Processing, Heraklion, Crete, Greece, pp. 328–331. IEEE (2016)
5. Havanki, B.: Moving Hadoop to the Cloud Harnessing Cloud Features and Flexibility for Hadoop Clusters. O'Reilly Media, Sebastopol (2017)
6. Astrova, I., Koschel, A., Lennart, M.H., Nahle, H.: Offering Hadoop as a cloud service. In: Proceedings of the 2016 SAI Computing Conference, London, UK, pp. 589–595. IEEE (2016)
7. Kalja, A., Reitsakas, A., Saard, N.: e-Government in Estonia: best practices. In: Anderson, T.R., Daim, T.U., Kocaoglu, D.F., Piscataway, N.J. (eds.) Technology Management: A Unifying Discipline for Melting the Boundaries. pp. 500–506. IEEE (2005)
8. Kalja, A., Robal, T., Vallner, U.: New generations of Estonian e-Government components. In: Proceedings of the 2015 PICMET, Portland, Oregon, USA, pp. 625–631. IEEE (2015)

Application of Graph Clustering and Visualisation Methods to Analysis of Biomolecular Data

Edgars Celms, Kārlis Čerāns, Kārlis Freivalds, Paulis Ķikusts, Lelde Lāce,
Gatis Melkus, Mārtiņš Opmanis, Dārta Rituma, Pēteris Ručevskis,
and Juris Vīksna(✉)

Institute of Mathematics and Computer Science, University of Latvia, Riga, Latvia
{edgars.celms,karlis.cerans,karlis.freivalds,paulis.kikusts,
lelde.lace,gatis.melkus,martins.opmanis,darta.rituma,
peteris.rucevskis,juris.viksna}@lumii.lv

Abstract. In this paper we present an approach based on integrated use of graph clustering and visualisation methods for semi-supervised discovery of biologically significant features in biomolecular data sets. We describe several clustering algorithms that have been custom designed for analysis of biomolecular data and feature an iterated two step approach involving initial computation of thresholds and other parameters used in clustering algorithms, which is followed by identification of connected graph components, and, if needed, by adjustment of clustering parameters for processing of individual subgraphs.

We demonstrate the applications of these algorithms to two concrete use cases: (1) analysis of protein coexpression in colorectal cancer cell lines; and (2) protein homology identification from, both sequence and structural similarity, data.

Keywords: Clustering algorithms · Graph visualization
Biomolecular networks · Bioinformatics

1 Introduction

The recent fast development of sequencing and other high-throughput experimental techniques for gathering biomolecular data has significantly contributed to the possibility to describe and analyse different biological processes in genome wide (i.e. whole organism) level, or even at the level of the all known biomolecules (e.g. proteins). Many of these biological processes can be described as networks (metabolic, signalling, gene regulation, protein homology etc.), i.e. as graphs with vertices representing biomolecules (genes, proteins, metabolites, different types or RNA etc.) and edges representing different types of interactions and/or relations between them. The information about such interactions or relations very often is not discrete (either by their nature, when edges between vertices only indicate some degree of similarity, or due to experimental limitations, which

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 243–257, 2018.
https://doi.org/10.1007/978-3-319-97571-9_20

allow to assign only probabilities to possible interactions), thus networks are described by (either directed or undirected) weighted complete graphs with number of vertices ranging from several thousands to tens of thousands. Thus, there is an obvious need of suitable and sufficiently efficient algorithms for analysis of such graphs and extracting from them biologically useful information.

Very often biological questions of interest can be (at least partially) formulated in terms of finding clusters in such networks satisfying certain properties. The problem of finding such 'biologically interesting' clusters is also the most widely studied from computer science perspective and it is also the focus of our study. More concretely, here we are considering two types of biomolecular networks where 'good clustering' of network vertices could lead of identification of biologically significant features: networks describing correlation of protein abundance expressions in cell lines, and networks describing protein homology in terms of both their sequence and their structure similarity.

There is enormous amount of clustering algorithms and lot of them are used also for graphs [21]. Regarding practical applications, network clustering is probably most frequently used for analysis of different types of social networks, nevertheless biological networks have been widely analysed ([4] provides a comprehensive overview of both, different clustering methods, and their applications specifically to social and biological networks.

Most of these methods could be applied also to biomolecular networks that have been obtained by high-throughput experiments, however the latter usually have certain distinctive features, making direct application of these methods difficult. One of these features is already mentioned high network connectivity – they tend to be described by complete graphs with edge weight assignments. Due to number of reasons there are usually no weight threshold values that could be uniformly applied to the whole network to obtain sparser graphs without loosing biologically significant edge connections. Often one also should expect that 'biologically good' clustering will contain clusters of very different sizes, so there is no optimal cluster size that should be achieved.

A comprehensive survey of use of clustering algorithms for analysis of biomolecular data generated by high-throughput experimental methods is given in [19]. The primary focus is on spanning tree based approaches, but the survey is largely descriptive without going into specificity of concrete biological questions that are being addressed. Another well-known method shown to be successful for biomolecular data is Markov Cluster (MCL) algorithm. In [24] the authors describe a custom-built implementation of MCL algorithm for biomolecular network analysis together with few application examples. However, from the biological perspective a much more detailed application use case has been presented already by [3]; the problem considered – identification of protein families with structural similarities – is also related to protein homology analysis we present here. Another well developed field of study is analysis of protein interaction networks in order to identify protein complexes (e.g. [15]); this problem is also partially related to correlation network analysis described here.

The emphasis, however, in most of these studies is comparison of results of clustering and other network analysis methods with already known biological facts and in measuring how similar the obtained results are (in particular the approach described in [3] largely just compares the outcomes of two computational methods, although the one is notably more complicated than another). But clearly, much more interesting would be development of methods, which by themselves could *discover* new currently unknown relations in biomolecular networks, which latter could be submitted to much more rigorous biological validation. Although few, there have already been several studies explicitly stating as a goal the discovery of new biological hypotheses by (mostly) computational techniques. In [14] the authors use IMHRC clustering method for discovery of new protein complexes and the approach has been largely successful. The specific network analysis problem, however, is comparatively simple, since already sparse manually annotated protein interaction networks are used as input. Biologically significant protein 'communities' have been automatically identified in [10] solely from high-throughput data, still new discoveries are made for a specific organism (rat) using already known extensive biological knowledge about similar organisms.

In our work we emphasise the possibility to use graph analysis techniques to generate new biological knowledge and demonstrate with the two presented use cases that this could be at least partially achieved. Without the claims to introduce fundamentally novel ideas for graph clustering, we describe three clustering algorithms targeted for analysis of concrete types of biological networks. Two of these algorithms address specific features of these networks that are not usually addressed by general purpose clustering methods. (These specific features are related, correspondingly, to the need to interpret differently edges showing high positive and high negative correlations, and to the need to apply clustering not to the single weighted network, but essentially to differences between two given weighted networks.) We also demonstrate the particular usefulness of semi-supervised (in particular, used in combination with graph visualisation methods for result assessment) approach for discovery of 'new knowledge' in biomolecular network generated by high-throughput experimental techniques.

For the described case studies we have used in-house graph visualisation software packages 'geocx' and SNA containing implementations of several complex graph layout algorithms and providing well-developed APIs for integration with other software components. These packages have been developed within several commercial projects, open source versions unfortunately have not been released, but parts of the software could be available on request. A number of the implemented graph layout algorithms (e.g. [5,6]) have been developed also by several authors of this paper. There are also several publications describing applications of this visualisation software to a number of practical problems ranging from analysis of social networks to biomolecular data analysis [12,20,25]. Both packages 'geocx' and SNA provide similar functionality, but have been developed for use correspondingly in C++ and Java programming environments. In this study SNA has been mostly used for analysis of protein abundance correlation networks and 'geocx' for protein homology analysis.

2 Analysis of Protein Abundance Correlation Networks

In this section we investigate clustering of networks that describe correlations between protein concentration levels in different cell lines. The possibility to experimentally measure protein concentrations at the whole genome level by high-throughput mass spectrometry experimental techniques is very recent, the technology is quite expensive and the number of published datasets is still very small. We have analysed proteomics data for COREAD project cell lines – one of the largest data sets available [2], containing measurements of concentrations of 6893 human proteins in 33 different colorectal cancer cell lines (the data are actually given for around 9000 proteins and 40 cell lines, part of which we have omitted mostly due to data incompleteness). The data set is available from Expression Atlas database [18] with accession number E-PROT-6. Correlation network was constructed by choosing 6893 proteins as vertices of complete graph in which edge between two proteins P_1 and P_2 has weight equal to Pearson correlation coefficient between concentration levels of these two proteins across all 33 cell lines (thus all edge weights are from the interval $[-1, 1]$).

Given such correlation network, a natural question to ask from biological perspective is whether there can be assigned biologically known relationships between similarly expressed genes – i.e. genes that correspond to particular clusters in network. The question was partially addressed already [2], and it was observed that there is increased correlation between genes from network clusters and either sets of proteins that form so-called protein complexes, or sets of proteins with common transcription factors (i.e. there is a common regulatory process for production of such proteins within a cell). The described study used R software package WGCNA [13], which is custom designed for such type of analysis.

These results, however, only show that proteins that belong to known 'biological clusters' (i.e. either their production is similarly regulated, or they are stable only as parts of specific complexes) tend to be better clustered together in correlation network, which can be measured by assigning a simple score (e.g. clustering coefficient or similar). From the computational perspective, however, much more interesting question is whether we can identify biological relationships between proteins by identifying and analysing clusters in correlation network. The particular data set is also very appropriate for such analysis, since in this case the methods of biological validation of any discovered candidate clusters are already known.

A particular feature of correlation graph is the presence of two types of edges with positive and negative weights that correspond to positive and negative correlations. A large absolute value of edge weight (i.e. close either to 1 or to -1) can indicate a likely biological relationship between proteins. At the same time, whilst it is natural to expect clusters with edges of large positive weights, by definition such strongly connected clusters can not be formed by negative weight edges. Thus, there is a need to treat these two types of edges separately. Whilst in principle clustering of graphs with both positive and negative weighted edges (known also as signed networks) has been studied before (see e.g. [23]), the analysis has mostly been applied to social networks, and, although certain restrictions on distribution of negative weight edges are usually assumed, these are weaker than restrictions needed for protein correlation networks.

One of fundamental clustering approaches is based on spanning trees and is widely used for biological data (e.g. [19]). The algorithm constructs the heaviest tree and chooses appropriate edge weights for thresholding. This approach gives good enough insight in the graph structure, however induced subgraphs of particular clusters can be weakly tied (see example graph in Fig. 1 with threshold 0.85).

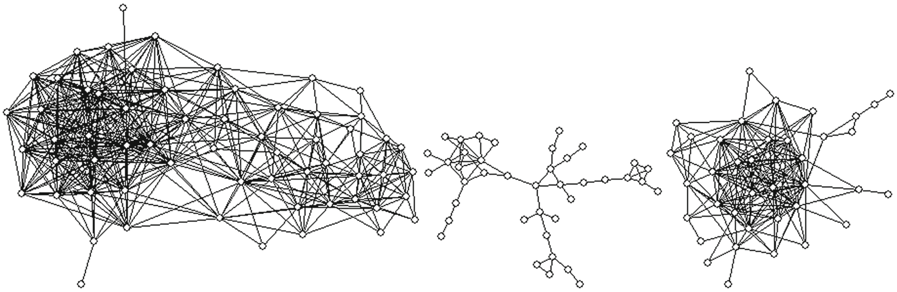


Fig. 1. A part of clustered protein correlation graph using B-MST heuristic

Another widely used method for analysis of clusters formed by positive weight edges is Markov Cluster (MCL) algorithm [24]. Unfortunately, experiments with MCL also show that using this algorithm for our data sets weakly tied clusters are unavoidable (Fig. 2).

This has motivated us to develop algorithm focused on finding more dense clusters reflecting mutually correlating objects. The algorithm also provides an additional option to choose subset of vertices to allow more deeply investigate parts of graph we are interested in. The pseudocode of the developed Ratio Based Augmenting Clustering algorithm is described in Algorithm 1.

Natural way of algorithm usage is choosing all G vertices as V_0 and considering only edges with positive weights.

By choosing $\rho = 1$ we obtain clustering based on cliques, i.e. $\mathcal{R}_i$ $(i \geq 2)$ are complete subgraphs with i vertices of G.

The time complexity of the described algorithm is $O(n^{i_{max}})$ where n is the number of G vertices. Although the time complexity of MCL algorithm is only $O(n^3)$, the exponential complexity of Algorithm 1 did not pose any practical difficulties.

We performed series of tests on a particular biomolecular data set comparing our approach – implementation of Algorithm 1 (V_0 - all vertices, $d = [0.85, 1.0]$, $i_{min} = 4$, $i_{max} = 6$, $\rho = 0.25$, execution time 10.66 s), and publicly available implementation of MCL algorithm ($threshold = 0.85$, $inflation = 1.4$, execution

Algorithm 1. RatioBasedAugmentingClustering

```
Input. Graph $G$. Range $d$. Subset $V_0$ of $G$ vertices. Indices $i_{min}, i_{max}$. Ratio $\rho$ $(0 < \rho \leq 1)$
Output. Clusters of $G$ vertices.

1. Build thresholded graph $G^t$ – subgraph of $G$ with edges having weights within the given range $d$.
2. Build lists $\mathcal{R}_2$,..., $\mathcal{R}_i$,..., $\mathcal{R}_{i_{max}}$ where $\mathcal{R}_i$ is the set of induced subgraphs with $i$ vertices of $G^t$. $\mathcal{R}_2$ elements are pairs of adjacent vertices where at least one vertex belongs to $V_0$. Every element of $\mathcal{R}_{i+1}$ $(i \geq 2)$ is obtained from some $\mathcal{R}_i$ element $r$ by adding every vertex $v$ having at least $max(2, \rho \cdot i)$ edges from $v$ to the vertices of $r$.
3. Build $G^t$ subgraph $G'$ comprising all $G^t$ vertices and all edges from $\mathcal{R}_j (j \geq i_{min})$.
4. Connected components of $G'$ are considered as clusters.
```

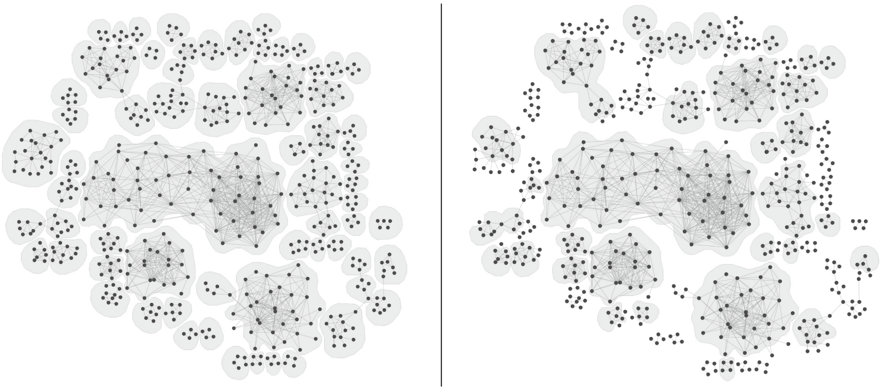


Fig. 2. Parts of protein correlation network processed correspondingly by MCL algorithm (left) and our Algorithm 1 (right). (To facilitate comparison, only edges within clusters identified by MCL algorithm are shown.) Although MCL identifies approximately two times more clusters, part of them are weakly tied and it was not possible to associate these with some biological meaning.

time 3.7 s). The comparison is shown in Fig. 2 (It should be noted that only parts of the graph are shown and small clusters are not presented).

The exact number and sizes of identified clusters varies with parameters used, but 36 of the identified clusters remained comparatively stable for range of parameter values, the sizes of these clusters varied between 4 and 60 proteins (with average cluster size around 10). The biological validity of these clusters were assessed by manual comparison with data in publicly available bioinformatics databases, and in the most cases there was a good correspondence either with published protein complexes, or proteins co-regulated by a common transcription factor. To 33 from 36 identified clusters it was possible to unambiguously associate them with specific complexes or co-regulated groups of proteins (the remaining 3 did not have any clear biological meaning), although some larger clusters contained several different protein complexes. The number of outliers (cluster elements not belonging to associated complexes) was small and in some cases their presence had biological explanation. However, only in few cases the identified clusters contained all the proteins from associated complexes, although this can be at least partially explained by missing data – E-PROT-6 dataset contains information about only half of the known human proteins and in addition 25% of its entries were excluded from analysis due to their incompleteness.

For analysis of clusters induced by negative weight edges we have developed a heuristic Star Clustering algorithm (Algorithm 2). The underlying assumption is that a high level of connectivity of a set of proteins by negative weight edges to one (or few) other proteins might be indicative of biological relationships between the proteins from such a set. Additional evidence could be provided by medium high connectivity with positive weighted edges within the protein set, however, the positive weight thresholds might be too low to identify such set as a cluster only on basis of them. The algorithm was applied to the same data set as Algorithm 1.

Algorithm 2. NegativeStarClustering

```
Input. Graph $G$. Range $d^-$. Range $d^+$. Degree $k$. Indices $i_{min}, i_{max}$. Ratio $\rho$ ($0 < \rho \leq 1$)
Output. Clusters of $G$ vertices.

1. Build thresholded graph $G^-$ – subgraph of the initial graph with edges having
   negative weights within the given range $d^-$.
2. Rank $G^-$ vertices according to their degrees. Let $S$ be set of vertices with degree
   at least $k$. Let $L$ be vertices incident in $G^-$ with $S$ and not belonging to $S$.
3. return RatioBasedAugmentingClustering($G$, $d^+$, $L$, $i_{min}$, $i_{max}$, $\rho$)
```

Range d^+ should be such that only positive edges are processed by RatioBasedAugmentingClustering. Time complexity of Algorithm 2 is the same as of Algorithm 1.

In Fig. 3 we show the final result of Algorithm 2 with parameters $d^- = [-1.0, -0.7]$, $d^+ = [0.85, 1.0]$, $k = 5$, $i_{min} = 4$, $i_{max} = 6$, $\rho = 0.25$, execution time 0.83 s. In this case 9 stable clusters Ri0−Ri8 have been identified.

A cursory examination (by expert in the filed) of the clusters obtained revealed that almost all of them roughly correspond to individual protein complexes registered in the UniProt database. How representative the clusters are of the complexes varies considerably, but no single cluster features all known members of its complex, and two of the nine clusters (Ri6, Ri7) feature representatives of two different complexes, albeit ones with closely related function and localization. Furthermore, we found that one particular cluster Ri3 featured a substantially more varied group of proteins associated with interferon-mediated immune response, which means that even in cases where the clustered proteins do not belong to distinct complexes the algorithm appears to successfully detect a functionally meaningful association between them.

Thus, in general it can be said that clusters identified on basis of negative correlations within network describe biologically related sets of proteins, although these biological relations tend to be more subtle than for clusters based on positive correlation weights.

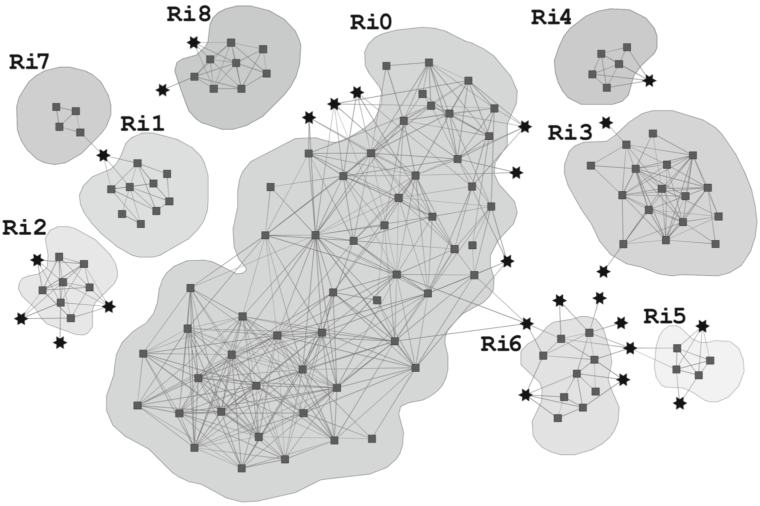


Fig. 3. Graph processed by Algorithm 2. Clusters are labeled: ribosomal proteins (Ri0), chromosomal passenger complex and other proteins (Ri1), COP9 signalosome (Ri2), interferon cluster (Ri3), actin-related protein complex (Ri4), replication factor C (Ri5), GINS and CHAF1 complexes plus other replication-related proteins (Ri6), DNA polymerase and DNA primase (Ri7), cytochrome C oxidase (Ri8)

3 Analysis of Protein Homology on Basis of Structure and Sequence Similarity

In this section we demonstrate the application of a combined visualisation-clustering approach for identification of new (i.e. biologically still unconfirmed) candidate sets of evolutionary related proteins. Here by combined approach we mean that the results of a number of intermediate steps of clustering algorithm are visualised as graphs at which points, if needed, it is possible to manually adjust several algorithm parameters (in this specific case mostly edge weight thresholds) in order to obtain the optimal results.

The biological problem studied in this section is the following.

Two or more proteins are said to be evolutionary related if they share a common ancestor, and if this is the case they are called *homologous*. From computational perspective proteins can be regarded as sequences in 20 letter alphabet and the simplest way to assess their similarity is to compute sequence similarity score e.g. by Smith-Waterman algorithm [22]. For two sequences of length n the algorithm runs in $O(n^2)$ time and gives the most precise results (according to the current theory on sequence evolution), although for large data base searches faster heuristic methods (such as BLAST) are often used.

Given similarity score of two proteins the usual assumption is that they are homologous for scores 30% or more and non-homologous for scores below 15% with a 'twilight zone' between these values (these thresholds are rough estimates and depend on many factors, including the exact method used for score computation). In this twilight zone homology can be more carefully assessed using multiple sequence comparison. The exact algorithms for this require exponential time ($O(n^k)$ for k sequences), but good heuristic alignment programs exist, notably ClustaW and Clustal Omega [8,9]. Still the running times for multiple alignments are such, that they can be used for analysis of selected pre-chosen protein sets and not for searching of homology relations in the set of all known protein sequences.

Biological functions of proteins, however, are better described by their 3D structures than sequences. Unfortunately, whilst it is assumed that in the most cases 3D structures are uniquely defined by sequences, there is no known method for computing structures from sequences and structures need to be determined experimentally. This is a much more difficult task than determination of sequences and the number of known protein structures is roughly 10 times lower than the number of known protein sequences.

Experimentally determined protein structures in already pre-analysed form are included in several protein classification data bases, one of the two most popular (and more bioinformatics-friendly) being CATH [16]. The main three classes of CATH classification contains domains of what are biologically known as α, β and α-β proteins (α and β are names for the main well-defined fragments of 3D structures). These classes are further hierarchically partitioned, the fourth level of this partitioning is called *homologous superfamily* and belonging of two proteins to the same or to two different superfamilies is generally considered a good indicator of protein homology or its absence (and in some cases sequence similarities within a homologous might be lower than 15% threshold).

Although not frequent, biologically-confirmed examples of homologous proteins with different 3D structures are known [7], and probabilities of such 'fold changes' have been analysed also by several authors of this paper [11,12,26].

Our main assumption for homology analysis presented here is that for larger clusters of proteins within which there are many linked protein pairs with sequence similarity above some threshold, the probability that such sequence similarities have occurred by chance is considerably lower than probability that the same similarity has occurred by chance just for a pair of proteins. Therefore detection of such clusters might be an indicator of evolutionary related protein groups. Assigning even and approximate probability score to a cluster described by a graph G with vertices corresponding to proteins and edges indicating similarity above some threshold t seems to be a very difficult combinatorial problem. However, low probabilities for 'loosely connected' clusters have been confirmed by simulation experiments using a toy model (assuming protein sequences from only 2 amino acids with sequence length $n \leq 10$ and considering graphs G with up to 7 vertices). The assumption is also consistent with very small number of sequences $P_1, \ldots, P_n$ of 4 or more real proteins with sequence similarities at least t for pairs (P_i, P_{i+1}) and lower than $t - \Delta t$ for all other pairs (with values of t from twilight zone interval $[15, 30]$ and $\Delta t = 1 \ldots 3$).

Such observation by itself is not very useful for homology detection if the only available information is a single matrix with similarity scores between proteins, since it already requires knowledge of subgraphs G describing putative homology clusters. However, the situation is much more promising if there are two or more matrices and/or graphs describing homology relations between proteins that are assigned by different methods – e.g. a matrix of sequence similarity scores and hierarchical clustering of proteins assigned by CATH classification. In such a case one can look for graphs G described by 'unexpected edges', i.e. edges between proteins that are treated as homologous only by one of the two methods, and, if there is a high level of connectivity between vertices, select them for more scrupulous homology assessment (e.g. by multiple sequence alignment). The approach is semi-automatic and relies on combined use of graph visualisation and graph analysis methods to detect sufficiently compact and well interconnected candidates G for putative homology clusters. In a more informal way a similar approach was used in [12] for detecting fold changes that can arise from small sequence mutations.

Graphs containing different types of edges are known as multilayer networks and have been extensively studied before, a comprehensive analysis of their properties and description of a range of practical applications is presented e.g. in [1]. Nevertheless, the previous work seems to be more focused on finding common features shared by network layers, and not on identification of unexpected significant differences between the layers.

Here we propose the following semi-interactive Algorithm 3 for identification of candidate homology clusters.

In *Step 1* T is the largest possible edge weight in G, in current implementation $T = 100$ corresponding to the largest possible sequence similarity of 100%. The first semi-interactive part of the algorithm is *Step 2*, where there is an option to select manually G^t with the 'best proportion' of two types of edges. The goal is to obtain graph with as many edges as possible within CATH homology superfamilies and as few as possible between different superfamilies. The best choice is decided on visualisation results and the optimal thresholds are slightly different for each of analysed CATH classes. In *Step 3* the degree threshold d is given as input to algorithm. The purpose is to minimise number of edges that need to be drawn by visualisation program, whilst still keeping highly connected parts compact and well separated. For concrete datasets that were used good results were obtained with $d = 3$. In *Step 4* the threshold t' is adjusted both heuristically and on basis of visualisation results (the used values fluctuated around 20) and the previously computed set of edges E'' are used here for guidance. In *Step 5* it is possible manually to select clusters with different t' thresholds. The supported

Algorithm 3. HomologyClusterIdentification

```
Input. Weighted labelled undirected graph $G = (P, W, L)$, where $P$ is set of proteins, $\mathcal{L}$ is set of CATH homology superfamilies, $W : P \times P \rightarrow \{0, \ldots, T\}$, $L : P \rightarrow \mathcal{L}$.
Output. Clusters of $G$ vertices.

1. For each $t = 0, \ldots, T$ compute unweighted graph $G^t = (P, E^t)$ with $\{p_1, p_2\} \in E^t$ iff $W(p_1, p_2) \geq t$.
2. Chose $G' = (P, E') = G^t$ with the 'best proportion' between edges with $L(p_1) = L(p_2)$ and edges with $L(p_1) \neq L(p_2)$.
3. For each $l \in \mathcal{L}$ define $G'_l$ as subgraph of $G'$ induced by set of vertices $P_l = \{p \in P \mid L(p) = l\}$. Using a greedy strategy, starting with pairs of vertices $p_1, p_2 \in P_l$ with the highest minimal degree of $p_i$ in graph $G'_l$ until there are no more pairs $p_1, p_2 \in P_l$ with minimal degree larger than $d$, remove edges $\{p_1, p_2\}$ both from $G'_l$ and $G'$. Let $G'' = (P, E'')$ be subgraph of $G'$ with edges that were not removed from $E'$.
4. Build $\hat{G} = (\hat{P}, \hat{E})$ by including in set $\hat{E}$ all edges $\{p_1, p_2\} \in E''$ with $L(p_1) \neq L(p_2)$ and $W(p_1, p_2) \geq t'$, and including in set $\hat{P}$ all vertices $p \in P$ that are endpoints at least of one edge from $\hat{E}$.
5. Output as clusters all $k$-edge-connected components of graph $\hat{G} = (\hat{P}, \hat{F})$.
```

values of cluster connectivity were $k = 1 \ldots 3$. A fragment of visualised graph used in cluster identification process is shown in Fig. 4.

The running time of the algorithm for graph is $O(Tn^2 + k\hat{n}^4)$, where n is initial number of vertices in G and $\hat{n}$ is number of vertices in the largest connected component of $\hat{G}$. For the analysis of proteins from CATH classes we have $T = 100$ and maximal value of $n \approx 10000$, and the first 4 steps required less than a minute on standard workstation. The running time $O(k\hat{n}^4)$ for *Step 5* corresponds to a straightforward adaptation of minimum cut algorithm. For our datasets the maximal value of $\hat{n}$ was less that 50 and the computations were very fast, but, if the algorithm is adapted for other similar problems, a more efficient approach for partitioning $\hat{G}$ into components might be needed.

For analysis we used the available CATH S95 representative set that contains only selected proteins with sequence similarity not exceeding 95%. Without losing any candidates for homology clusters, this allows to work with much smaller graphs that can still can be processed by visualisation tools. Classes 1, 2 and 3 of α, β and α–β were analysed separately and contained correspondingly 4679, 5668 and 10626 protein domains. For each CATH class sequence similarities between all pairs of proteins $\{P_1, P_2\}$ were computed by *ssearch* implementation of Smith-Waterman algorithm [17], and a normalised percentage similarity within range $0 \ldots 100\%$ was computed as $ssearch\_score(P_1, P_2)/min\{length(P_1), length(P_2)\}$ (such percentage score assignment is well-motivated mathematically and is known to work well for comparatively short CATH domains, although numerically scores tend to be lower compared to assignments computed by BLAST or other heuristic methods).

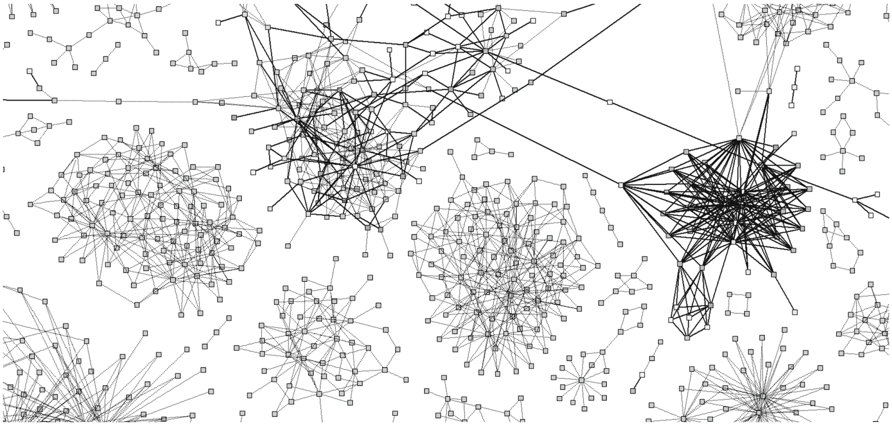


Fig. 4. A fragment of CATH 2 homology network. The edges between vertices of identified candidate clusters are shown with bold lines, dotted lines connect proteins within known clusters, i.e. proteins with similar sequences that belong to the same homologous superfamily. From the latter type only a subset E'' remaining after *Step 3* of Algorithm 3 is shown

The number and sizes of identified candidates of homology clusters depends on the exact parameters used, however, we have identified 16 stable (i.e. not significantly changing for different parameter values) and well connected clusters with 5 or more proteins – clusters with sizes $6, 6, 8, 11, 12, 14$ in CATH 1, clusters with sizes $5, 18, 28, 30$ in CATH 2 and clusters with sizes $5, 8, 9, 9, 25, 41$ in CATH 3. A sample cluster (actually consisting of two clusters with good sequence alignments within each of them) from CATH 2 is shown in Fig. 5.

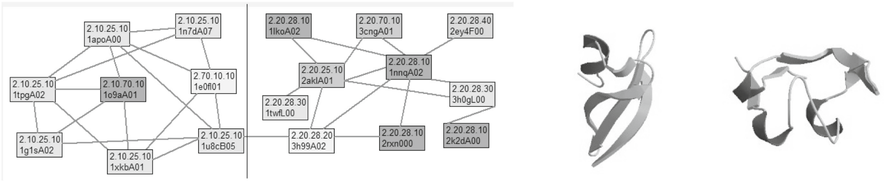
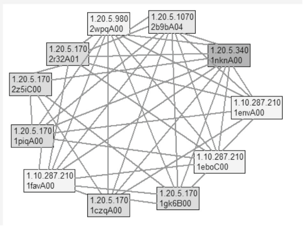


Fig. 5. A two component cluster from CATH 2. Each of components consists of sequences with comparatively good multiple alignments. The cluster on the right includes proteins from 2 notably different fold groups: 2.20.25 and 2.20.28

These candidate clusters were closer inspected by multiple sequence alignments and some of them can be regarded as artefacts, e.g. both largest clusters from CATH 2 show increased sequence similarity due to protein modification by adding polyhistadine tags – sequences of 6 or more amino acids that are added to all the proteins from these clusters. Similarly explained can be 3 smaller CATH

3 clusters, however, the remaining 11 clusters do not contain artificially modified proteins and good sequence alignments suggest that similarity of proteins within these clusters might have biological reasons. One of these clusters from CATH 1 containing 11 protein domains together with part of their sequence alignment is shown in Fig. 6.



```
----------MDFNPSEVASQVTNYIQAIAAAGVGVLALAIGLSAAWKYAKRFLKG----
VGALTVLFGAIAYGEVTAAAATGDAAAVQEAAVSAILGLIILL--GINLGLVA-------
--------------PESPKGP--DILVVLLSVMGAILLIGLAALLIWKLLITIHDRKEF-
-------------GALEERAI--PIWWVLVGVLGGLLLLTILVLAMWKVGFFKRNRP---
--------GCPAEQRASPLTS------IISAVV-GIL-LVVVL--GVVFGILIKRRQQKI
-----------EFQTLSPEGS--GNLAVIGGVAVGVVLLLVLA--G--VGFFIHRRRK--
-------------GPLVPRGS--MALIVLGGVAG----LLLFI--G--LGIFFSVRSRHR
--------GVIDTSAVESAIT--DGQGDMKAIGGYIVGALVIL-AVAGLIYSMLRKA---
-----SGGGGVDVGDVVSAIQ--GAAGPIAAIGGAVLTVMVGI-----KVYKWVRRAM--
-------MQSVITD-VTGQLT--AVQADITTIGGAI----IVL-AAVVLGIRWIKAQFF-
-------AERSLSGLTEEEAI--AVHDQFKTTFSAF----IIL-AAVAHVLVWVWKPWF-
```

Fig. 6. A candidate cluster of homologous proteins form CATH 1 and (part of) sequence alignment computed by Clustal Omega

4 Conclusions

It can be said that in both use cases of biomolecular networks analysis that we have presented here the obtained results are promising. In analysis of network describing correlations of protein expressions across different cell lines we have shown that it is possible to identify biologically related groups of proteins solely by computational means from experimental data obtained by high-throughput methods (which, in particular, means that no pre-existing biological knowledge has already been included in design of experiments). The biological validity of the discovered clusters have also been confirmed. Whilst the correspondence between automatically identified clusters of proteins and known biologically related protein groups was not exact, it should be noted that practically there were no false-positives (i.e. no 'good' clusters were discovered for which biological relationships have not been found).

In protein homology analysis we have identified 17 candidate clusters of homologous proteins, which, as far as we know, have not been detected by other biological or bioinformatics methods. Whilst for 5 of these clusters it was found that the reasons for their existence are artificial (protein modifications by adding sequence tags), this actually could be interpreted as a positive result - i.e. the method allowed to identify a weak signal that had a biological (albeit artificial) explanation. This allows to expect that the remaining 12 clusters might also have biological (and probably more natural) explanations. In this case, however, there are no obvious experimental methods that could be used for their biological validation. Our hypothesis is that these clusters could characterise some aspects related to protein structure evolution (which currently still is not well understood).

From the perspective of computer science, we have presented 3 new heuristic clustering algorithms for dense graphs. Whilst these do not introduce fundamentally novel ideas for graph clustering, Algorithm 1 has shown better performance than MCL algorithm that is widely used on similar type of data. Algorithms 2 and 3 address some specific features of biomolecular networks that are not usually addressed by general purpose clustering methods – correspondingly, the need to interpret differently edges showing high positive and high negative correlations, and the need to apply clustering not to the single weighted network, but essentially to differences between two given weighted networks. We anticipate that both these methods could be adapted for other practical applications and likely merit further development.

Acknowledgements. The research was supported by ERDF project 1.1.1.1/16/A/135.

References

1. Boccaletti, S., et al.: The structure and dynamics of multilayer networks. Phys. Rep. **544**, 1–122 (2014)
2. Choudhari, J., et al.: Genomic determinants of protein abundance variation in colorectal cancer cells. Cell Rep. **20**, 2201–2214 (2017)
3. Enright, A., et al.: An efficient algorithm for large-scale detection of protein families. Nucleic Acids Res. **30**, 1575–1584 (2002)
4. Fortunato, A.: Community detection in graphs. Phys. Rep. **486**, 75–174 (2010)
5. Freivalds, K., Dogrusoz, U., Kikusts, P.: Disconnected graph layout and the polyomino packing approach. In: Mutzel, P., Jünger, M., Leipert, S. (eds.) GD 2001. LNCS, vol. 2265, pp. 378–391. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45848-4_30
6. Freivalds, K., Glagoļevs, J.: Graph compact orthogonal layout algorithm. In: Fouilhoux, P., Gouveia, L.E.N., Mahjoub, A.R., Paschos, V.T. (eds.) ISCO 2014. LNCS, vol. 8596, pp. 255–266. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-09174-7_22
7. Grishin, N.: Fold change in evolution of protein structures. Struct. Biol. **134**, 167–185 (2001)
8. Higgins, D., Sievers, F.: Clustal Omega, accurate alignment of very large numbers of sequences. Methods Mol. Biol. **1079**, 105–116 (2014)
9. Higgins, D., et al.: ClustalW and ClustalX version 2.0. Bioinformatics **23**, 2947–2948 (2007)
10. Jonsson, P., et al.: Cluster analysis of networks generated through homology: automatic identification of important protein communities involved in cancer metastasis. BMC Bioinform. **7**(1), 2 (2006)
11. Kurbatova, N., Mancinska, L., Viksna, J.: Protein structure comparison based on fold evolution. Lect. Notes Inform. **115**, 78–89 (2007)
12. Kurbatova, N., Viksna, J.: Exploration of evolutionary relations between protein structures. Commun. Comput. Inf. Sci. **13**, 154–166 (2008)
13. Langfelder, P., Horwath, S.: WGCNA: an R package for weighted correlation network analysis. BMC Bioinform. **9**, 559 (2008)
14. Maddi, A., Eslahchi, C.: Discovering overlapped protein complexes from weighted PPI networks by removing inter-module hubs. Sci. Rep. **7**, 3247 (2017)
15. Nepusz, T., Yu, H., Paccanaro, A.: Detecting overlapping protein complexes in protein-protein interaction networks. Nat. Methods **9**, 471–472 (2012)

16. Orengo, C., et al.: New functional families in CATH to improve the mapping of conserved functional sites to 3D structures. Nucleic Acids Res. **44**, 490–498 (2013)
17. Pearson, R.: Effective protein sequence comparison. Methods Enzymol. **266**, 227–258 (1996)
18. Petryszak, R., et al.: Expression Atlas update - an integrated database of gene and protein expression in humans, animals and plants. Nucleic Acids Res. **44**(D1), 746–752 (2016)
19. Pirim, H., Eksioglu, B., Perkins, A.: Clustering high throughput biological data with B-MST, a minimum spanning tree based heuristic. Comput. Biol. Med. **62**, 94–102 (2015)
20. Rung, J., Schlitt, T., Brazma, A., Freivalds, K., Vilo, J.: Building and analysing genome-wide gene disruption networks. Bioinformatics **18**, S202–S210 (2002)
21. Schaeffer, S.: Graph clustering. Comput. Sci. Rev. **1**, 27–64 (2007)
22. Smith, T., Waterman, M.: Identification of common molecular subsequences. J. Mol. Biol. **147**, 195–197 (1981)
23. Traag, A., Doreian, P., Mrvar, A.: Partitioning signed networks. ArXiv e-prints abs/1803.02082 (2018)
24. van Dongen, S., Abreu-Goodger, C.: Using MCL to extract clusters from networks. In: van Helden, J., Toussaint, A., Thieffry, D. (eds.) Bacterial Molecular Networks. Methods in Molecular Biology (Methods and Protocols), vol. 804, pp. 281–295. Springer, New York (2012). https://doi.org/10.1007/978-1-61779-361-5_15
25. Vihrovs, J., Prusis, K., Freivalds, K., Rucevskis, P., Krebs, V.: A potential field function for overlapping point set and graph cluster visualization. Commun. Comput. Inf. Sci. **550**, 136–152 (2015)
26. Viksna, J., Gilbert, D.: Assessment of the probabilities for evolutionary structural changes in protein folds. Bioinformatics **23**, 832–841 (2007)

A New Knowledge-Transmission Based Horizontal Collaborative Fuzzy Clustering Algorithm for Unequal-Length Time Series

Shurong Jiang, Jianlong Wang, and Fusheng Yu(✉)

School of Mathematical Sciences, Beijing Normal University,
Laboratory of Mathematics and Complex Systems, Ministry of Education,
Beijing 100875, China
yufusheng@bnu.edu.cn

Abstract. This paper focuses on the clustering of unequal-length time series which appear frequently in reality. How to deal with the unequal lengths is the key step in the clustering process. In this paper, we will change the given unequal-length clustering problem into several equal-length clustering sub-problems by dividing the unequal-length time series into equal-length time series. For each sub-problem, we can use the standard fuzzy c-means algorithm to get the clustering result which is represented by a partition matrix and a set of cluster centers. In order to obtain the final clustering result of the original clustering problem, we will use the horizontal collaborative fuzzy clustering algorithm to fuse the clustering results of these sub-problems. In the process of collaboration, the collaborative knowledge is transmitted by partition matrixes whose sizes should be the same. But in the scenario here, the obtained partition matrixes most often have different sizes, thus we cannot directly use the horizontal collaborative fuzzy clustering algorithm. Taking into account the collaborative mechanism of the horizontal collaborative fuzzy clustering algorithm, this paper here presents a novel method for extending the partition matrixes to have same sizes. This method can make the partition knowledge be effectively transmitted and thus assume the good final clustering results. Experiments showed the effectiveness of the proposed method.

Keywords: Horizontal collaborative fuzzy clustering
Unequal-length time series · Knowledge transmission · Partition matrix

1 Introduction

Time series pervades the fields of meteorology, economics, biomedicine, communication engineering and so on, such as hourly temperature change in a certain area, the daily closing index change of stock market, the daily visits of web pages, the monthly GDP changes in a certain area, and the sales volume of stores. In reality, due to the loss of data in some time periods and the confidentiality of data, or the different start times of collecting data, etc., the lengths of time series are often not equal. How to finish the clustering a set of such kind of time series is a problem we face with. It is clear that we cannot directly use the standard fuzzy c-means (FCM) algorithm. In order to solve the

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 258–271, 2018.
https://doi.org/10.1007/978-3-319-97571-9_21

clustering problem of time series with unequal lengths, this paper presents a new knowledge-transmission based horizontal collaborative fuzzy clustering (HCFC) algorithm for unequal- length time series data.

The HCFC algorithm was proposed by Pedrycz originally [1]. It aims at exploring clusters of a given data set. In HCFC algorithm, a group of patterns are described in different feature spaces resulting in different datasets. The standard FCM algorithm is implemented in each dataset. Then, the clustering results of all datasets are collaborated to obtain the structure of the given patterns by taking into account all the datasets. The intensity of collaboration can be different. This depends on the dataset involved and the purpose of collaboration. Since the emergence of HCFC, a lot of related algorithms have been proposed. According to formalizations of the objective function appeared in these algorithms, they can be divided into two categories: (1) global objective-based HCFC [2–6], and (2) local objective-based HCFC [7–13].

In the process of collaboration of HCFC, the collaboration is implemented by transmitting the clustering knowledge presented by the partition matrix from the collaborating dataset to the collaborated dataset. Since the same patterns and the cluster numbers to be produced, the sizes of all the partition matrixes obtained on all datasets are the same. Thus, when we use the HCFC algorithm to do the clustering of unequal-length time series, we should first segment the unequal-length time series into several parts. In each part, all the subsequences have the same length. Therefore, the standard FCM algorithm can be carried out on each part to obtain a partition matrix. But the sizes of the partition matrixes of different parts are different, because different parts have different numbers of subsequences. This makes it impossible for HCFC algorithm to be carried out directly. This paper proposes a new method for extending the partition matrixes from unequal-sizes to equal-sizes. In details, we use the last partition matrix which has biggest size to extend the other partition matrixes to have the same sizes. In this way, the HCFC algorithm can be carried on. Experiments show that the proposed extending method is effective, and this extending method based HCFC algorithm is of high efficiency.

2 Preliminaries

In this section, we briefly recall HCFC, and introduce a partition index employed in this paper.

2.1 Review of the HCFC

Given N patterns, if they are described in different feature spaces, then there will resulted in different datasets. Let X[j] be the dataset of the N patterns described in a m [j]-dimension feature space ($j = 1, 2, \ldots, K$). The purpose of the HCFC algorithm is to find the structure among the given K patterns with considering all the datasets corresponding to different feature spaces. Instead of carrying out the standard fuzzy c-means (FCM) clustering algorithm on the biggest dataset composed of all of the K smaller datasets, the HCFC algorithm carries out a supervised clustering on one smaller dataset by fusing all the clustering results of the other smaller datasets produced by the standard FCM algorithm. The clustering result of each smaller dataset is transmitted by the partition matrix. The clustering process exhibits a distinct collaboration feature between one small collaborated dataset and other smaller collaborating datasets. The strength of each collaborating dataset on the collaborated dataset can be quantified by a parameter.

Let X[j] be the collaborated dataset, and the others be the collaborating datasets. $\alpha(j,p)$ denotes the collaboration strength between X[j] and X[p] (p $\neq$ j). Then the HCFC can be implemented by the following objective function:

$$\min Q[j] = \sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^2[j]\, d_{ik}^2[j] + \sum_{p\neq j}^{K} \alpha[j,p] \sum_{i=1}^{c}\sum_{k=1}^{N} (u_{ik}[j] - u_{ik}[p]\,)^2 d_{ik}^2[j]$$

$$s.t \sum_{i=1}^{c} u_{ik}[j] = 1\,(j = 1,2,\cdots,K; k = 1,2,\cdots,N)$$

Where $U[j] = (u_{ik}[j])_{c\times N}$ is the partition matrix of dataset $X[j]$, $d_{ik}^2[j]$ is the distance between the k-th pattern $X_k[j]$ and the i-th prototype $v_i[j]$ $(j = 1,2,\cdots,K;\ i = 1,2,\cdots,$ $c;\ k = 1,2,\cdots N)$, c is the number of the clusters to be produced.

The first term in the objective function is the same to the one of the standard FCM algorithm when it is applied to the local data set $X[j]$. The second term quantifies the collaboration between the collaborated dataset and the collaborating datasets.

From the second term of the objective function, it can be found that all the partition matrixes should have the same size. Otherwise, the collaboration cannot be implemented.

By the Lagrange multiplier method, we have

$$u_{st}[j] = \frac{\varphi_{st}[j]}{1+\psi[j]} + \frac{1-\sum\limits_{i=1}^{c}\varphi_{it}[j]\Big/(1+\psi[j])}{\sum\limits_{i=1}^{c} d_{st}^2[j]\Big/ d_{it}^2[j]}, \qquad v_{sh}[j] = \frac{A_{sh}[j] + C_{sh}[j]}{B_s[j] + D_s[j]},$$

$$(s = 1,2,\cdots,c; t = 1,2,\cdots,N; h = 1,2,\cdots,n[j])$$

where, $\varphi_{st}[j] = \sum\limits_{p\neq j}^{K} \alpha[j,p] u_{st}[j]$, $\quad \psi[j] = \sum\limits_{p\neq j}^{K} \alpha[j,p]$

$$A_{sh}[j] = \sum_{k=1}^{N} u_{sk}^2 x_{kh}, \qquad C_{sh}[j] = \sum_{p\neq j}^{K} \alpha[j,p] \sum_{k=1}^{N} (u_{sk}[j] - u_{sk}[p])^2 x_{kh},$$

$$B_s[j] = \sum_{k=1}^{N} u_{sk}^2[j], \qquad D_s[j] = \sum_{p\neq j}^{K} \alpha[j,p] \sum_{k=1}^{N} (u_{sk}[j] - u_{sk}[p])^2.$$

2.2 Partition Index

The partition index of a partition matrix, $P(U)$, is defined as follows [14]:

$$P(U) = \frac{1}{N}\sum_{i=1}^{c}\sum_{k=1}^{N} u_{ik}^{2}$$

When $u_{ik} = 1/c(i = 1, \cdots, c; k = 1, \cdots, N)$, the value of $P(U)$ is $1/c$, which is the minimum; When all instances belong to only one cluster, the value of $P(U)$ is 1, which is the maximum. The closer the value of $P(U)$ is to 1, the better the result of clustering is.

3 The New Knowledge-Transmission Based Horizontal Collaborative Fuzzy Clustering Algorithm for Unequal-Length Time Series

In this section, we propose a new method of knowledge-transmission in collaborative fuzzy clustering for unequal-length time series.

Subsection 3.1 gives the clustering idea based HCFC algorithm for unequal-length time series. Subsection 3.2 presents the new knowledge-transmission mechanism. Subsection 3.3 shows the proposed algorithm.

3.1 The Idea for Clustering for Unequal-Length Time Series

In order to using the HCFC algorithm in the clustering of unequal-length time series, we sort the unequal-length time series firstly. Next, we segment the unequal- length time series into several parts. In each part, all the subsequences have the same length. Thus, the standard FCM algorithm can be implemented to obtain the partition matrix. Because the lengths of the original time series are different, the sizes of the partition matrixes may be different. We use the partition matrix of the last part to extend the others to have the same size. Then, the HCFC algorithm can be carried out. The clustering result of the last part collaborated by the other parts can be regarded as the clustering result of the original unequal-length time series.

Let $X = \{x_1, x_2, \cdots, x_N\}$ be a set of N time series, where $x_i = \left\{ \left(t_1^{(i)}, x_1^{(i)}\right), \left(t_2^{(i)}, x_2^{(i)}\right), \cdots, \left(t_{l_i}^{(i)}, x_{l_i}^{(i)}\right) \right\}(i = 1, 2, \cdots, N)$, with $t_j^{(i)} < t_{j+1}^{(i)}$ and $x_j^{(i)} \in R$, $(j = 1, 2, \cdots, l_i)$. l_i is the length of time series x_i. There exists $i \neq j$ to make $l_i \neq l_j$. That is, there are at least two time series with different lengths.

First, the time series are sorted, ensure: $\left|x_1'\right| \geq \left|x_2'\right| \geq \ldots \geq \left|x_1'\right| \geq \ldots \geq \left|x_N'\right|$. x_1' is the i-th time series after sorting. $\left|x_i^{'}\right|$ is the length of time series $x_i^{'}$ (see Fig. 1). For the sake of convenience, we still use x_i to denote the i-th time series after sorting. Let m be the length of the longest time series x_1. Then, the time series after sorting are segmented into K parts. In every part, all the subsequences have the same length. After that, we get K parts of subsequences: $Part_1, Part_2, \cdots, Part_K$ (see Fig. 2).

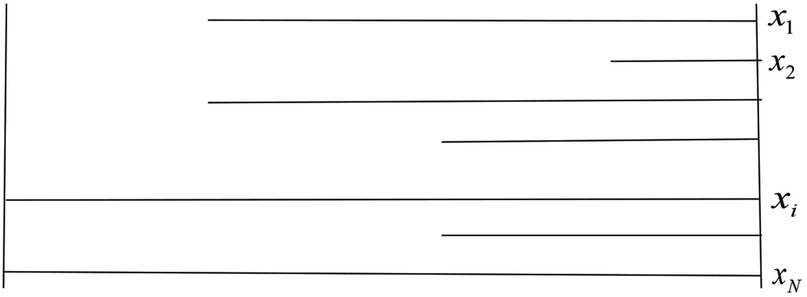


Fig. 1. The unequal-length time series before sorting and segmenting

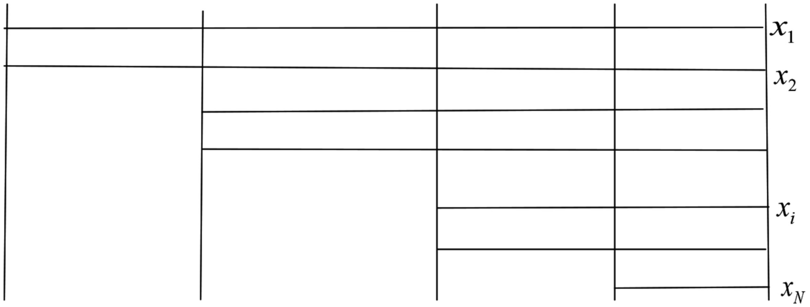


Fig. 2. The unequal-length time series after sorting and segmenting

Let M_1 be the number of subsequences in the first part, $M_1 + M_2 + \cdots + M_j$ is the number of subsequences in the j-th part $Part_j$ $(j = 2, 3, \cdots, K)$. Let m_j be the length of $Part_j$ $(j = 1, 2, \cdots, K)$. Then, we have

$$Part_i \cap Part_j = \phi \ (i \neq j), \quad \cup_{i=1}^{K} Part_i = X,$$

$$m_1 + m_2 + \cdots + m_K = m$$

The subsequences of every part have the same length after segmenting. Suppose $Part_j$ is to be clustered into c_j clusters, s.t. $c_1 \leq c_2 \leq \cdots \leq c_j \leq \cdots \leq c_K$. By the standard FCM algorithm, we can obtain the partition matrix $U_0[j]$ of $Part_j (j = 1, 2, \cdots, K)$, whose size is $\left(\sum_{i=1}^{j} M_j \right) \times c_j$. The sizes of the K partition matrixes are different. So, the HCFC algorithm cannot be carried out directly. In the nest subsection, we will propose a new method for extending the partition matrixes to have the same size.

3.2 The New Knowledge-Transmission Mechanism

It is assumed that the original time series in $X = \{x_1, x_2, \cdots, x_N\}$ are clustered into c clusters. Let $U = \{u_{ik}\}_{c \times N}$ be the partition matrix of X, our aim is to extend the size of partition matrixes $U_0[j]$ from $\left(\sum_{i=1}^{j} M_j\right) \times c_j$ to $c \times N\,(j = 1, 2, \cdots, K-1)$. The extended partition matrix is denoted as $U[j] = \{u_{ik}[j]\}_{c \times N}$. In literature [15], $U_0[j]$ was extended in the following manner:

$$U[j]\,(i,k) = \begin{cases} U_0[j]\,(i,k),\ 1 \le i \le c_j,\ \le k \le \sum_{i=1}^{j} M_i \\ 0,\ \ others \end{cases}$$

The manner of extending is illustrated in Fig. 3

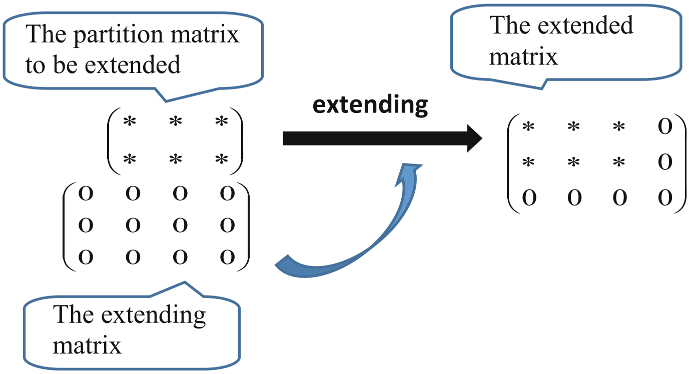


Fig. 3. Illustration of the old manner of extending partition matrix

In this extending manner, $U_0[j]$ is extended by 0. In the process of collaboration, the entries of $U = \{u_{ik}\}_{c \times N}$ corresponding to the ones in $U[j]$ extended by 0 will forced to approach to 0. But the 0's are of no meaning, such an approach will also influent other entries of $U = \{u_{ik}\}_{c \times N}$. In order to avoid of the unnecessary influence, we present a new extended manner here.

In our new method, $U_0[j]$ is extended by the partition matrix $U_0[K]$ of the last part $Part_K$. This is implemented by Eq. (1) and the manner of the extending is illustrated in Fig. 4:

$$U[j]\,(i,k) = \begin{cases} U_0[j]\,(i,k),\ 1 \le i \le c_j,\ 1 \le k \le \sum_{i=1}^{j} M_i \\ U_0[K]\,(i,k),\ others \end{cases} \tag{1}$$

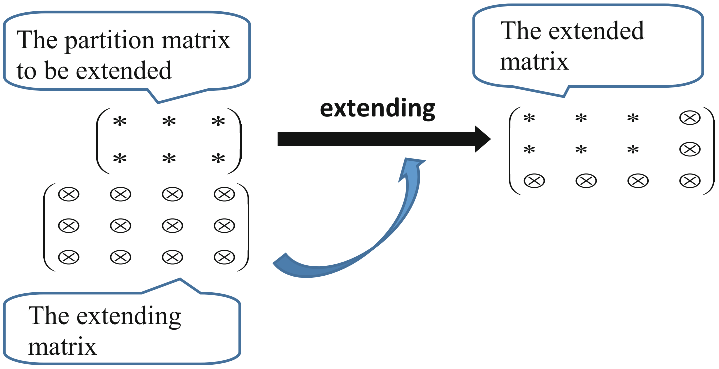


Fig. 4. Illustration of new manner of extending partition matrix

3.3 The Algorithm of the Proposed Method

The clustering algorithm of our proposed method based on the new extending manner is described by the following steps:

Step 1: Sorting and segmenting of the given unequal-length time series: Order the set of unequal length time series of X to ensure $|x_1'| \geq |x_2'| \geq \ldots \geq |x_1'| \geq \ldots \geq |x_N'|$. Then segment the ordered time series into K parts: $Part_1, Part_2, \cdots, Part_K$. The number of subsequences and the length of $Part_j$ are $\sum_{i=1}^{j} M_i$ and m_j respectively.

Step 2: Using the standard FCM algorithm to obtain the partition matrix $U_0[j]$ of $Part_j$, where the number of clusters to be produced is $c_j\ (j = 1, 2, \cdots, K)$ satisfying $c_1 \leq c_2 \leq \cdots \leq c_j \leq \cdots \leq c_K$.

Step 3: Using $U_0[K]$ to extend partition matrixes $U_0[j](j = 1, 2, \cdots, K-1)$ according to formula (1).

Step 4: Carrying out the HCFC algorithm given in Sect. 2 on the last part $Part_K$ with the following two added steps:

(1) Initializing partition matrix U with $U_0[K]$.
(2) Using the extended matrixes obtained in Step 3 to be the collaborative matrixes.

4 Experimental Evaluation

In this section, we design two experiments to show the performance of the new proposed algorithm. In each experiment, a comparison between the proposed method and the old method is made.

4.1 Experiment on the Synthetic Control Dataset

The aim of this experiment is to testify the effectiveness of the proposed algorithm.

The Synthetic Control Dataset

The experimental data $X = \{x_1, x_2, \cdots, x_{12}\}$ is selected from Synthetic Control Dataset in UCI dataset, each of which is a time series. It has three clusters (see Table 1). The first four time series have the same length 60, the next four time series have the same length 40, while the last four time series have the same length 20.

Table 1. Synthetic Control Dataset in UCI dataset

Cluster name	Clusters
Cluster 1	$\{x_1, x_5, x_8\}$
Cluster 2	$\{x_2, x_4, x_7\}$
Cluster 3	$\{x_3, x_6, x_9, x_{10}, x_{11}, x_{12}\}$

Clustering Result

These 12 unequal-length time series are segmented into three parts: $Part_1, Part_2$ and $Part_3$ who have 4, 8, and 12 subsequences respectively. All the subsequences in each part have the same length (see Fig. 5). Let $c_1 = 2$, $c_2 = c_3 = 3$ be the numbers of clusters to be produced by FCM on the three parts respectively. So, the sizes of the partition matrixes $U_0[1]$, $U_0[2]$ and $U_0[3]$ are 4×2, 8×3 and 12×3 respectively. Let $\alpha[1] = 0.15$, $\alpha[2] = 0.85$. Do the collaborative fuzzy clustering on $Part_3$ collaborated by $Part_1$ and $Part_2$ with the proposed method. Figure 6 shows the membership functions of the three clusters produced.

By the principle of maximum membership, each time series is clustered to one cluster (see Table 2). From Fig. 6, we can find that: Time series x_1, x_5 and x_8 are clustered in Cluster 1, time series x_2, x_4, and x_7 are clustered in Cluster 2, while time series x_3, x_6, x_9, x_{10}, x_{11} and x_{12} are clustered in Cluster 3. This clustering result is consistent with the actual structure of the given dataset.

It should be emphasized that the membership degree of time series 12 to Cluster 2 is smaller than 0.5, while to Cluster 3 is bigger than 0.5. Thus, by the principle of maximum membership, it should be cluster to Cluster 3.

Table 2. The clustering result produced by the proposed algorithm

Cluster name	Clusters produced by the proposed algorithm
Cluster 1	$\{x_1, x_5, x_8\}$
Cluster 2	$\{x_2, x_4, x_7\}$
Cluster 3	$\{x_3, x_6, x_9, x_{10}, x_{11}, x_{12}\}$

Comparison

Do the collaborative fuzzy clustering on *Part*$_3$ collaborated by *Part*$_1$ and *Part*$_2$ with the old method under the same initializations. Figure 7 shows the membership functions of the three clusters produced. By the principle of maximum membership, we can obtain

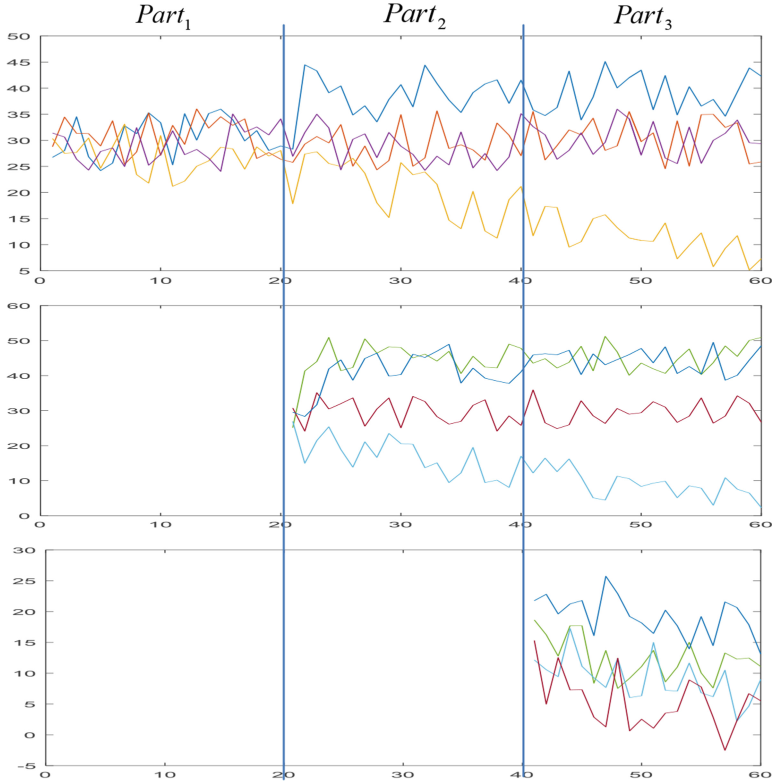


Fig. 5. Partial synthetic control data of time series with unequal lengths

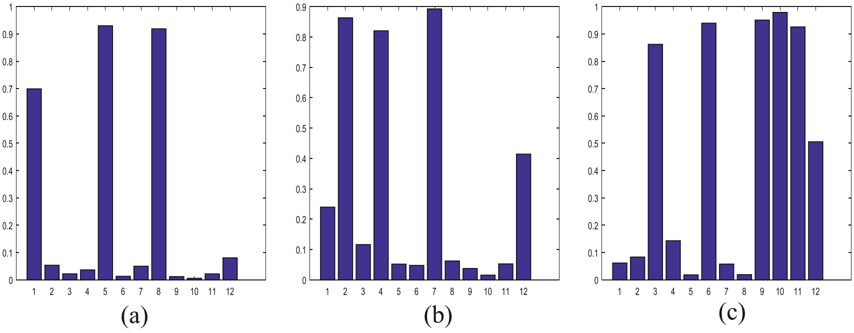


Fig. 6. The membership functions of the three clusters produced by the proposed algorithm: (a) the first cluster; (b) the second cluster; (c) the third cluster

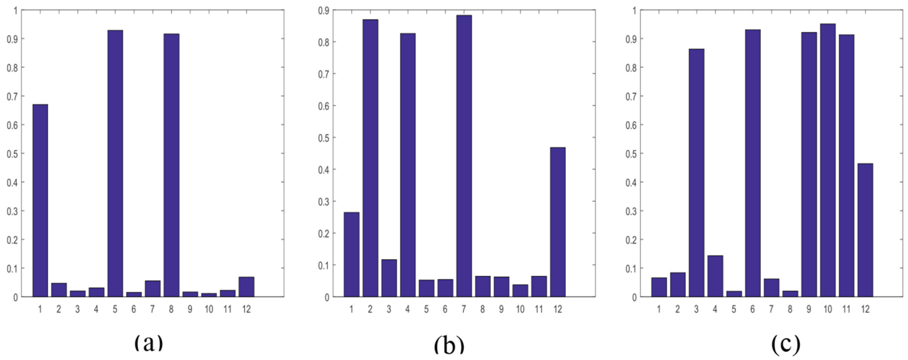


Fig. 7. The membership functions of the three clusters produced by the old algorithm: (a) the first cluster; (b) the second cluster; (c) the third cluster

the hard partition of the given dataset. Table 2 shows the comparison of the clustering results of the proposed algorithm and the old algorithm.

From Table 3, we can see that: only time series 12 belongs to different clusters in the two algorithms. In the result of the old algorithm, the membership degree of time series 12 to Cluster 2 is 0.4679, while to Cluster 3 is 0.4638. Thus, by the principle of maximum membership, it should be clustered to Cluster 2. In fact, time series x_{12} has a downward tendency. It should belong to Cluster 3 where every time series presents a downward tendency. While in the result of the proposed algorithm, time series x_{12} belongs to the right cluster. This exhibits the better performance of the proposed algorithm than the old algorithm.

Table 3. The comparison of the clustering results of the proposed method and the old method

Cluster name	Clusters produced by the proposed algorithm	Clusters produced by the old algorithm
Cluster 1	$\{x_1, x_5, x_8\}$	$\{x_1, x_5, x_8\}$
Cluster 2	$\{x_2, x_4, x_7\}$	$\{x_2, x_4, x_7, x_{12}\}$
Cluster 3	$\{x_3, x_6, x_9, x_{10}, x_{11}, x_{12}\}$	$\{x_3, x_6, x_9, x_{10}, x_{11}\}$

4.2 Experiment on the UCR Time Series

In this experiment, a comparison between the proposed algorithm and the old algorithm is made in both partition index and time consuming.

Dataset

The experimental dataset $X = \{x_1, x_2, \cdots, x_{18}\}$ is selected from UCR time series. Table 4 summarizes the selected time series. This dataset has 3 clusters, namely Coffee, Car and MALLAT. The same length of the first six time series is 150. The same length of the second six time series is 100. While the same length of the last six time series is 50.

Table 4. UCR time series selected for experimental studies

Cluster name	Clusters
Coffee	$\{x_1, x_2, x_3, x_4, x_7, x_8, x_9\}$
Car	$\{x_6, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17}, x_{18}\}$
MALLAT	$\{x_5, x_{10}, x_{11}\}$

Clustering Result

These unequal-length time series are segmented into three parts: $Part_1, Part_2$ and $Part_3$ (see Fig. 8). Let $c_1 = 2$, $c_2 = c_3 = 3$ be the numbers of clusters to be produced by FCM on the three parts respectively. So, the sizes of the partition matrixes $U_0[1]$, $U_0[2]$ and $U_0[3]$ are 6 × 2, 12 × 3 and 18 × 3 respectively. Let $\alpha[1] = 0.15$, $\alpha[2] = 0.85$. Set the maximum number of iterations to be 1000 and the threshold be 0.0001, Do the collaborative fuzzy clustering on $Part_3$ collaborated by $Part_1$ and $Part_2$ with the proposed algorithm. Figure 9 shows the membership functions of the three clusters produced.

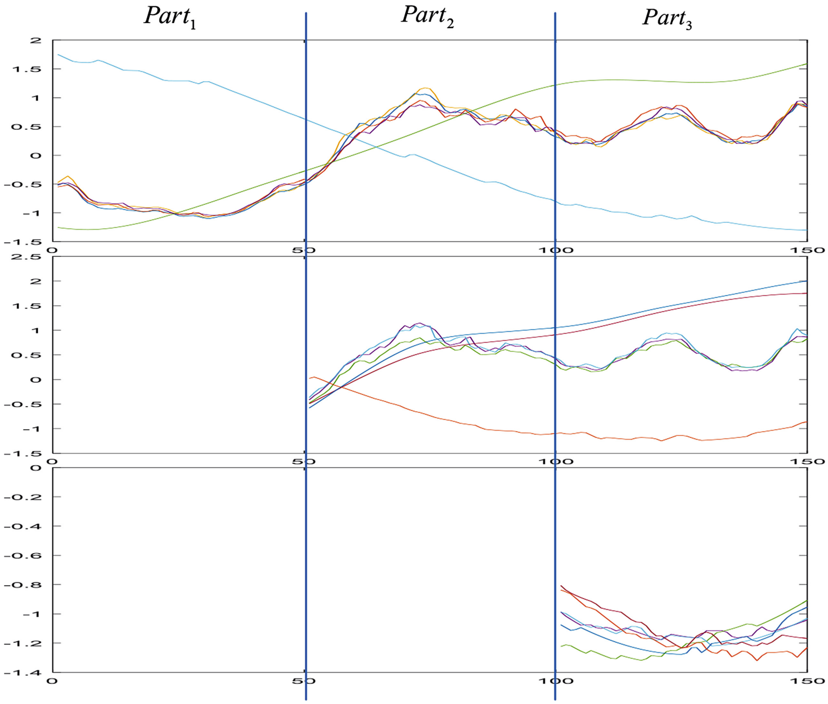


Fig. 8. Partial UCR time series of unequal lengths

According to the principle of maximum membership, from Fig. 9, we can see that x_1, x_2, x_3, x_4, x_7, x_8 and x_9 belong to the first cluster, x_6, x_{12}, x_{13}, x_{14}, x_{15}, x_{16}, x_{17} and

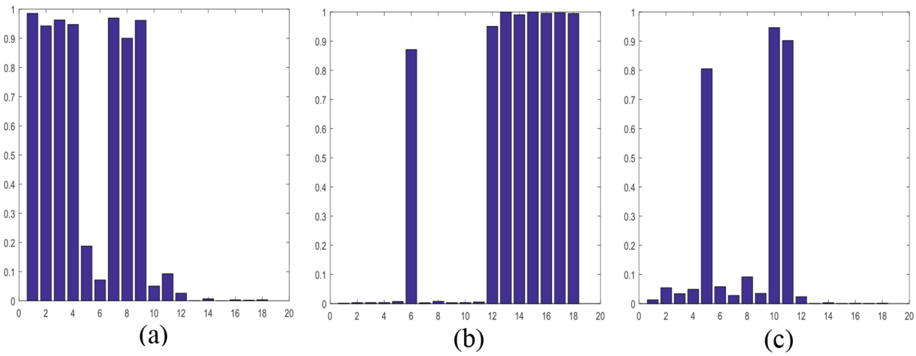


Fig. 9. The membership functions of the three clusters produced by the proposed algorithm: (a) the first cluster; (b) the second cluster; (c) the third cluster

x_{18} belong to the second cluster, x_5, x_{10} and x_{11} belong to the third cluster. The clustering result obtained is consistent with the actual structure of the given dataset (see Table 3).

Comparison

Do the collaborative fuzzy clustering on $Part_3$ collaborated by $Part_1$ and $Part_2$ with the old algorithm under the same initializations. Figure 10 shows the membership functions of the three clusters produced. We compare the two methods with respect to the value of $P(U)$ and the time efficiency (see Table 5).

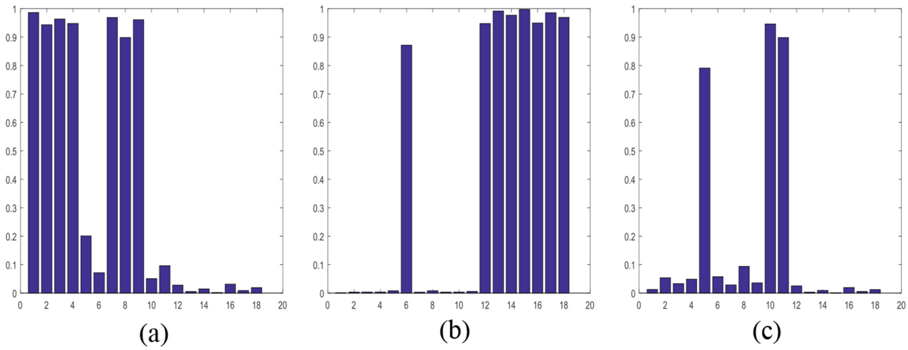


Fig. 10. The membership functions of the three clusters produced by the old algorithm: (a) the first cluster; (b) the second cluster; (c) the third cluster

From Table 5, we can see that:

- The value of $P(U)$ of the proposed algorithm is bigger than that of the old algorithm. It means that the clustering result of the proposed algorithm is clearer.
- The total time used by the proposed algorithm is 0.0427 s with. While the total time used by the previous algorithm is 0.0529 s. The proposed algorithm saves 19.3% of time. By contrast, we can draw a conclusion that the proposed algorithm has higher efficiency.

Table 5. The comparison of the two algorithms with respect to $P(U)$ and time consuming

Index	The proposed algorithm	The old algorithm
$P(U)$	0.9115	0.8980
Time (second)	0.0427	0.0529

5 Conclusion

Unequal-length time series can be seen often in reality. How to implement the clustering of unequal-length time series is of importance as well as difficulty. This paper presents this problem a HCFC-based method with a new knowledge-transmission mechanism. For some given unequal-length time series, they are first segmented into several parts. In each part, the lengths of all the subsequences are the same. We carry out the standard FCM algorithm on each part. After that, the clustering results of these parts are fused with the HCFC algorithm. In the process of HCFC algorithm, knowledge transmission cannot be transmitted directly due to the different sizes of the partition matrixes. In this paper, we propose a new manner to extend the other partition matrixes to have the same size. Then, the HCFC algorithm are carried out on the last part collaborated by the other parts. The first experiment shows the effectiveness of the algorithm of the proposed method. The second experiment shows that the algorithm of the proposed method not only gives better clustering result, but also reduces time consumption.

Acknowledgment. This work is supported by the National Natural Science Foundation of China (No. 11571001, No. 11701338), the Fundamental Research Funds for the Central Universities, Natural Science Foundation of Shandong Province (ZR2016AP12), and a Project of Shandong Province Higher Educational Science and Technology Program (J17KB124).

References

1. Pedrycz, W.: Collaborative fuzzy clustering. Pattern Recognit. Lett. **23**, 1675–1686 (2002)
2. Jiang, Y., Chung, F., Wang, S., Deng, Z., Wang, J., Qian, P.: Collaborative fuzzy clustering from multiple weighted views. IEEE Trans. Cybern. **45**, 688–701 (2015)
3. De Carvalho, F.d.A., Lechevallier, Y., De Melo, F.M.: Relational partitioning fuzzy clustering algorithms based on multiple dissimilarity matrices. Fuzzy Sets Syst. **215**, 1–28 (2013)
4. De Carvalho, F.d.A., De Melo, F.M., Lechevallier, Y.: A multi-view relational fuzzy c-medoid vectors clustering algorithm. Neurocomputing **163**, 115–123 (2015)
5. Zhou, J., Chen, C.L.P., Chen, L., Li, H.: A collaborative fuzzy clustering algorithm in distributed network environments. IEEE Trans. Fuzzy Syst. **22**, 1443–1456 (2014)
6. Cleuziou, G., Exbrayat, M., Martin, L., Sublemontier, J.: CoFKM: a centralized method for multiple-view clustering. In: Proceedings of the 9th IEEE Inter-national Conference on Data Mining, pp. 752–757 (2009)
7. Loia, V., Pedrycz, W., Senatore, S.: Semantic web content analysis: a study in proximity-based collaborative clustering. IEEE Trans. Fuzzy Syst. **15**, 1294–1312 (2007)

8. Prasad, M., Chou, K.P., Saxena, A., Kawrtiya, O.P., Li, D.L., Lin, C.T.: Collaborative fuzzy rule learning for Mamdanitype fuzzy inference system with mapping of cluster centers. In: Proceedings of the 2014 IEEE Symposium on Computational Intelligence in Control and Automation (2015)
9. Chou, K.P., Prasad, M., Lin, Y.Y., Joshi, S., Lin, C.T., Chang, J.Y.: Takagi–Sugeno–Kangtype collaborative fuzzy rule based system. In: Proceedings of the 2014 IEEE Symposium on Computational Intelligence and Data Mining, pp. 315–320 (2014)
10. Lin, C., Prasad, M., Chang, J.: Designing Mamdanitype fuzzy rule using a collaborative FCM scheme. In: Proceedings of the International Conference on Fuzzy Theory and Its Applications, pp. 279–282 (2013)
11. Prasad, M., Lin, C., Yang, C., Saxena, A.: Vertical collaborative fuzzy C-means for multiple EEG data sets. In: Proceedings of the Intelligent Robotics and Applications 6th International Conference, pp. 246–257 (2013)
12. Prasad, M., Lin, Y.Y., Lin, C.T., Er, M.J., Prasad, O.K.: A new data-driven neural fuzzy system with collaborative fuzzy clustering mechanism. Neurocomputing **167**, 558–568 (2015)
13. Han, Z., Zhao, J., Liu, Q., Wang, W.: Granular-computing based hybrid collaborative fuzzy clustering for long-term prediction of multiple gas holders levels. Inf. Sci. **330**, 175–185 (2016)
14. Hubert, L., Arabie, P.: Comparing partitions. J. Classif. **2**(1), 193–218 (1985)
15. Wang, X.: Intelligent Clustering and Forecasting of Large-scale Temporal Data. Doctoral thesis of Beijing Normal University (2013)

Non-index Based Skyline Analysis on High Dimensional Data with Uncertain Dimensions

Nurul Husna Mohd Saad(✉), Hamidah Ibrahim, Fatimah Sidi, and Razali Yaakob

Faculty of Computer Science and Information Technology,
Universiti Putra Malaysia, Selangor, Malaysia
nhusna.saad@gmail.com, {hamidah.ibrahim,fatimah,razaliy}@upm.edu.my

Abstract. The notion of skyline query is to find a set of objects that is not dominated by any other objects. Regrettably, existing works lack on how to conduct skyline queries on high dimensional uncertain data with objects represented as continuous ranges and exact values, which in this paper is referred to as *uncertain dimensions*. Hence, in this paper we define skyline queries over data with *uncertain dimensions* and propose an algorithm, *SkyQUD*, to efficiently answer skyline queries. The *SkyQUD* algorithm determines skyline objects through three methods that guaranteed the probability of each object being in the final skyline results: *exact domination*, *range domination*, and *uncertain domination*. The algorithm has been validated through extensive experiments employing real and synthetic datasets. Results exhibit our proposed algorithm is efficient and scalable in answering skyline query on high dimensional and large datasets with *uncertain dimensions*.

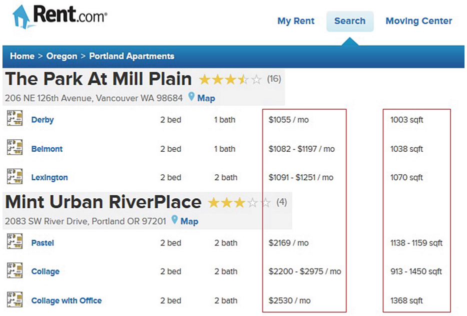
Keywords: Skyline query · Uncertain data · Uncertain dimensions

1 Introduction

Skyline retrieval paradigm has received a lot of attention since a decade ago as it proved especially useful for query personalisation. However, with the increasing number of applications that generate uncertain data, a significant amount of research has been committed for efficient skyline computation on uncertain data (e.g. [1,3,9,14,17]), yet, these existing works lack on how to conduct skyline queries on uncertain data with objects represented as continuous ranges and exact values. This form of uncertainty in data hereinafter shall be referred to as *uncertain dimensions*.

Uncertainty in data is not surprising as it can arise in a variety of scenarios such as in job listings and rental listings. For example in web rental listings, a property investor would like to do analysis for insights into the rental market. A search on *Rent.com* web page (www.rent.com) is performed and the search results are as illustrated in Fig. 1. Some of the property owners prefer to set the

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 272–286, 2018.
https://doi.org/10.1007/978-3-319-97571-9_22



The Park At Mill Plain ★★★½☆ (16)
206 NE 126th Avenue, Vancouver WA 98684 Map

Derby	2 bed	1 bath	$1055 / mo	1003 sqft
Belmont	2 bed	1 bath	$1082 - $1197 / mo	1038 sqft
Lexington	2 bed	2 bath	$1091 - $1251 / mo	1070 sqft

Mint Urban RiverPlace ★★★☆☆ (4)
2083 SW River Drive, Portland OR 97201 Map

Pastel	2 bed	2 bath	$2169 / mo	1138 - 1159 sqft
Collage	2 bed	2 bath	$2200 - $2975 / mo	913 - 1450 sqft
Collage with Office	2 bed	2 bath	$2530 / mo	1368 sqft

Fig. 1. Search results on rentals from *Rent.com*

price of rent within a range to indicate that they are willing to negotiate the rent, while other property owners prefer to give a fixed rent in order to keep future negotiations simple and straightforward.

When the values of one or more attributes are imprecise, it becomes more complicated to compute skyline on the uncertain data than on certain data. To the best of our knowledge, the study in [11] is the only work that has endeavored to tackle the issue of skyline query on data with *uncertain dimensions* by presenting an algorithm (*BBIS*), which performs the dominance testing between objects by comparing their median values (center points) and is indexed by an R*-tree structure. Despite the contributions made in [11], *BBIS* has its shortcoming when performing on high dimensionality dataset. This is largely due to the poor performance of the R*-tree index structure as it is well known that R*-tree could not adequately indexed large data of more than 5 dimensions [2,11,13]. Therefore, we make the following contributions in this paper:

1. We model the problem of computing skyline on *uncertain dimensions*.
2. We define the dominance relation and skyline on *uncertain dimensions*.
3. We propose an efficient non-index based algorithm for computing skyline probabilities (*SkyQUD*) that would gradually compute skyline probabilities on *uncertain dimensions* through three methods, namely: *exact domination*, *range domination*, and *uncertain domination*, that guaranteed the probability of each object being in the final skyline results.

The remainder of the paper is organised as follows. In Sect. 2, we provide the definitions and notations that are related to skyline queries on *uncertain dimensions*. In Sect. 3, we develop the *SkyQUD* algorithm for computing skyline probabilities on *uncertain dimensions*. We review the related works in Sect. 4 and we report the performance study conducted on *SkyQUD* algorithm in Sect. 5. In Sect. 6 we conclude the paper.

2 Preliminaries

The main idea in determining the candidates of skyline objects is to capture the dominant relationship between a set of objects.

Definition 1 (Dominance relation). *For two d-dimensional objects* $v = (v_1, v_2, \ldots, v_d)$ *and* $w = (w_1, w_2, \ldots, w_d)$, *v is said to dominate w (formally written as* $v \prec w$, *where it is assumed less is much preferred) if* $\forall i \in \{1, 2, \ldots, d\}$, $v_i \leq w_i$ *and* $\exists j \in \{1, 2, \ldots, d\}$, $v_j < w_j$.

Table 1. Running example for determining houses for rent that will be most preferred by users

ID	Rent	Sq. ft.	Bed	Bath	ID	Rent	Sq. ft.	Bed	Bath
a	1316-1908	534-639	1	1	*i*	1200	444-451	1	1
b	1556-1616	661-690	1	1	*j*	1505-2907	690	1	1
c	1660	644	1	1	*k*	1153	818	1	1
d	1845	907	1	1	*l*	1400	726	1	1
e	1850-2200	937	1	1	*m*	1136-1196	692	1	1
g	1407	1234-1363	2	2	*n*	920-980	409-455	1	1
h	1457	876-878	2	1	*o*	1415	865	1	1

This concept of dominance relation is used in conceptualising the skyline query [4–6,10,12].

Definition 2 (Skyline object). *Given a set of objects* $\mathbb{O}$, *an object* $v \in \mathbb{O}$ *is a skyline object if there exists no other object* $w \in \mathbb{O}$ *that dominates v. The skyline on* $\mathbb{O}$ *is the set of all skyline objects.*

Definition 3 (Continuous range). *Let* $[v_{lb} : v_{ub}]$ *denotes the continuous range of v such that* $v_{lb} \leq v \leq v_{ub}$ *and* $\{v_{lb}, v_{ub} \in \mathbb{R} | 0 \leq v_{lb} \leq v_{ub}\}$.

Consider Table 1 where the dataset represents a small sample which has been extracted from *Rent.com* database (www.rent.com). The rents of objects *a*, *b*, *e*, *j*, *m* and *n* are considered uncertain as they are represented as continuous ranges. Values that are in the form of continuous ranges will definitely induce a probability density function (pdf) over the ranges capturing the likelihood of possible values. Thus, the probability that *v* is in the continuous range $[v_{lb} : v_{ub}]$ can be computed by integrating its pdf over the range [15]:

$$Pr\left(v_{lb} \leq v \leq v_{ub}\right) = \int_{v_{lb}}^{v_{ub}} f\left(v\right) dv \tag{1}$$

The pdf $f(x)$ of an object x can be represented by various probability distributions depending on the type of distribution of the object. Following common methodology in the literature, we employed uniform probability distribution on all objects with continuous ranges for simplicity and ease of description.

The probability of an object with continuous range being in the skyline result set is the probability of that object not being dominated by any other objects. Following this, given the nature of the continuous range, the result of skyline query performed on objects with continuous range is bound to be probabilistic, since each object with continuous range is now associated with a probability value of it being a query answer. Now let us extend the previous concepts when *uncertain dimension* is introduced into the dataset.

Definition 4 (*Uncertain dimensions*). *Given a d-dimensional dataset* $\mathbb{D} = (D_1, D_2, \ldots, D_d)$ *with n objects. A dimension* $D_i \in \mathbb{D}$*, where* $(1 \leq i \leq d)$*, is said to be uncertain (denoted* $\mathbb{U}(D_i)$*) if it has at least one value* $a_j \in \mathbb{U}(D_i)$*, where* $(1 \leq j \leq n)$*, that is represented as a continuous range (denoted* $[a_{lb} : a_{ub}]_j$*); otherwise it is considered as a certain dimension. The continuous range is modeled as a probability density function defined on the real range* $[a_{lb} : a_{ub}]_j$ *where* a_{lb} *and* a_{ub} *are the lower-bound and upper-bound values of object a, respectively.*

The preceding definition assumes that *uncertain dimensions* are dimensions that contain at least one value that is represented as continuous range. Following this, given the nature of continuous range, the result of skyline query performed on objects with continuous range is bound to be probabilistic, in which, every object whose probability to not be dominated by any other objects is not zero would be included and reported as skyline objects, and thus making the size of skyline objects nearly equals to the size of the dataset in all scenarios. Therefore, a probability threshold τ is employed in the pruning process of our algorithm in order to manage the quality and the size of skyline objects reported. This means that the algorithm only accepts objects with a probability of at least τ to be in the skyline result set, and thus reducing the time users have to spend on making further analysis on the objects reported. Therefore, the issue that is the focus in this paper can be formally defined as:

Problem definition. *Given a set of n objects in a d-dimensional dataset* $\mathbb{D}$ *containing at least one uncertain dimensions* $\mathbb{U}(D_i)$*,* $(1 \leq i \leq d)$ *and for each object, the value in each dimension* $\mathbb{U}(D_i)$ *can take after the form of either exact value or continuous range with a probability density function defined over the range. Find the probabilistic skyline of the objects that satisfies the threshold* τ*.*

3 Skyline Probabilities of Objects in *Uncertain Dimensions*

In this section, we model skyline probabilities computation on *uncertain dimensions* and propose *SkyQUD* (Skyline Query on *Uncertain Dimensions*) algorithm that would gradually compute skyline probabilities on *uncertain dimensions*. To demonstrate the *SkyQUD* algorithm, we use the running example in Table 1. For ease of description, we represent the dimensions in Table 1 by D_1, D_2, D_3, and D_4, respectively. Given a d-dimensional dataset with *uncertain dimensions* as illustrated in Fig. 2. Thus, the dimensions in Table 1 are defined

	$\mathbb{C}$				$\mathbb{UC}$			
	D_1	D_2	...	D_α	$D_{\alpha+1}$	$D_{\alpha+2}$	...	D_d
v	5	3	$\cdots$	2	[110 : 120]	85	$\cdots$	[120 : 160]
w	4	4	$\cdots$	3	78	[60 : 85]	$\cdots$	[100 : 145]

Fig. 2. A d-dimensional dataset with *uncertain dimensions*

as $\mathbb{UC} = \{D_1, D_2\}$ and $\mathbb{C} = \{D_3, D_4\}$. The *SkyQUD* algorithm determines skyline objects by first categorising the objects into different groups before skyline dominance relations are performed. To partition the objects into distinctive groups, each object is examined to determine the existence of *uncertain dimensions*. Each object will have a corresponding list (denoted Θ) that will keep track of the *uncertain dimensions* that exist in a particular object. Once all objects have been examined and the corresponding lists have been obtained, the objects will then be grouped together according to the list Θ. By partitioning the objects into distinctive groups, unnecessary probability computations can be avoided. The number of distinctive groups (denoted ϱ) created varies depending on the number of *uncertain dimensions* $\mathbb{U}(D_i)$ that exists in a dataset, and the maximum number of possible distinctive groups is $\varrho \leq 2^{|\mathbb{UC}|}$, where $\mathbb{UC}$ is the set of *uncertain dimensions* in the dataset. Following this, by partitioning all objects in Table 1 would yield four groups of objects, that are $G_1 = \{c, d, k, l, o\}$, $G_2 = \{e, j, m\}$, $G_3 = \{g, h, i\}$, and $G_4 = \{a, b, n\}$.

Exact Domination. Having different groups of objects with different representations, different dominance relation techniques and skyline probability computations are needed to cater each group as discussed below. If a group consists of objects that are presented as an exact value in all dimensions, then the conventional dominance testing as defined in Definition 1 is sufficient enough to be implemented to this group since it is such a straightforward method without having to take into account the problem of continuous ranges. Continuing from the set of groups created previously, only G_1 qualifies for the traditional dominance test, and we would find that $c \prec d$ and $k \prec o$. On the other hand c, k, and l do not dominate each other as they are incomparable since $\forall i \in (2 \leq i \leq 4)$, $c.D_i \leq l.D_i \leq k.D_i$ and $k.D_1 < l.D_1 < c.D_1$. Thus, d will be pruned out and only c, k, and l will be the skyline candidates of G_1.

Range Domination. On the other hand, if a group consists of objects that are presented as a continuous range in at least one of the dimensions, it cannot be said with definite that an object totally dominates any other objects based solely on the dominance relation theory defined in Definition 1. To deal with such probabilities, for any two objects v and w with continuous ranges $[v_{lb} : v_{ub}]$ and $[w_{lb} : w_{ub}]$, respectively, we define seven possible types of relations between v and w. The probability of v to dominate w (denoted as $Pr(v < w)$) can be computed based on these relations following the probability theory as follows:

Definition 5 (Range-range value relations).
Disjoint: If $w_{lb} \geq v_{ub}$
$$Pr\,(v < w) = 1 \tag{2}$$
Disjoint-inverse: If $w_{ub} \leq v_{lb}$
$$Pr\,(v < w) = 0 \tag{3}$$
Overlap: If $v_{lb} \leq w_{lb} \leq v_{ub} \leq w_{ub}$
$$Pr(v < w) = 1 - \frac{1}{2}(Pr\{w_{lb} \leq v \leq v_{ub}\} \times Pr\{w_{lb} \leq w \leq v_{ub}\}) \tag{4}$$
Overlap-inverse: If $w_{lb} \leq v_{lb} \leq w_{ub} \leq v_{ub}$
$$Pr(v < w) = \frac{1}{2}(Pr\{v_{lb} \leq v \leq w_{ub}\} \times Pr\{v_{lb} \leq w \leq w_{ub}\}) \tag{5}$$
Contain: If $v_{lb} \leq w_{lb} \leq w_{ub} \leq v_{ub}$
$$Pr(v < w) = Pr\{v_{lb} \leq v \leq w_{lb}\} + \frac{1}{2}(Pr\{w_{lb} \leq v \leq w_{ub}\}) \tag{6}$$
Contain-inverse: If $w_{lb} \leq v_{lb} \leq v_{ub} \leq w_{ub}$
$$Pr(v < w) = Pr\{v_{ub} \leq w \leq w_{ub}\} + \frac{1}{2}(Pr\{v_{lb} \leq w \leq v_{ub}\}) \tag{7}$$
Equals: If $v_{lb} = w_{lb}$ *and* $v_{ub} = w_{ub}$
$$Pr(v < w) = \frac{1}{2}(Pr\{v_{lb} \leq v \leq v_{ub}\} \times Pr\{w_{lb} \leq w \leq w_{ub}\}) \tag{8}$$

The dominance relations between objects when there exist multiple $\mathbb{U}(D_k)$ can be explained as follows:

Definition 6 (Dominance relation on *uncertain dimensions*). *Given a d-dimensional dataset* $\mathbb{D} = \{\mathbb{A}, \mathbb{O}\}$*, where* $\mathbb{A}$ *represents a set of dimensions with different formats, i.e.* $\mathbb{A} = \{\mathbb{C}, \mathbb{UC}\}$*, and* $\mathbb{O}$ *represents a set of objects in the dataset. Formally,* $\mathbb{C}$ *consists of dimensions with exact values, i.e.* $\mathbb{C} = \{D_1, D_2, \ldots, D_\alpha\}$ *while* $\mathbb{UC}$ *consists of dimensions with continuous ranges and/or exact values, i.e.* $\mathbb{UC} = \{D_{\alpha+1}, D_{\alpha+2}, \ldots, D_d\}$*, as demonstrated in Fig. 2. An object* $v \in \mathbb{O}$ *is said to dominate another object* $w \in \mathbb{O}$ *(formally written as* $v \prec w$*, where it is assumed less is much preferred) if 1) in* $\mathbb{C}$*, where* $\forall i \in \{1, 2, \ldots, \alpha\}, v_i \leq w_i$ *and* $\exists j \in \{1, 2, \ldots, \alpha\}, v_j < w_j$*, and 2) in* $\mathbb{UC}$*, where* $\forall k \in \{\alpha+1, \alpha+2, \ldots, d\}, (1 - Pr(v_k < w_k)) < \tau$*.*

Thus, we define our skyline object for *uncertain dimensions* to be as follows:

Definition 7 (Skyline object on *uncertain dimensions*). *Given a d-dimensional dataset* $\mathbb{D} = \{\mathbb{A}, \mathbb{O}\}$*, defined in such a way that* $\mathbb{A} = \{\mathbb{C}, \mathbb{UC}\}$*,* $\mathbb{C} = \{D_1, D_2, \ldots, D_\alpha\}$*,* $\mathbb{UC} = \{D_{\alpha+1}, D_{\alpha+2}, \ldots, D_d\}$*, and* $\{v, w\} \in \mathbb{O}$*. Object v is a skyline object if there does not exist object w such that (1) w dominates v with certainty in* $\mathbb{C}$*, and (2) the probability of v to dominate w in* $\mathbb{UC}$ *is less than a probability threshold* τ*. That is, the skyline query on uncertain dimensions retrieves objects that satisfy* $\{v \in \mathbb{O} | \nexists w \in \mathbb{O}, w_{\mathbb{C}} \prec v_{\mathbb{C}} \wedge Pr(v_{\mathbb{UC}} < w_{\mathbb{UC}}) < \tau\}$*.*

Continuing from the above example, the dominance test in Definition 6 is applied separately on G_2, G_3, and G_4. For G_2, between objects e and j, since $j \prec e$ in $\mathbb{C}$ we can conclude that j is a possible skyline candidate. Therefore following Definition 7, for e to be a skyline candidate, e has to have the probability to not be dominated by j that is at least τ in $\mathbb{UC}$. We obtain $Pr(e.D_1 < j.D_1) = 0.63$ (following (7)) which would mean overall e is not dominated by j. However, m dominates e in $\mathbb{C}$ and $\mathbb{UC}$, thus removing e from further computations. Between m and j although $j \prec m$ in $\mathbb{C}$, $m \prec j$ in $\mathbb{UC}$, which would mean both objects are incomparable and thus making both objects as skyline candidates of G_2. The same method is applied on G_3 and G_4, where g, h, a, and b are filtered out from their respective groups making i and n as skyline candidate of G_3 and G_4, respectively.

Uncertain Domination. The objects that survived the filtering process of their own group in the previous steps are now considered as the skyline candidates. These objects however have to go through another filtering process, where they now will be compared to different groups in order to be finally accepted as skyline objects. To keep the comparisons between groups of local skyline candidates simple *SkyQUD* will treat group G_1 as a set of initial global skyline candidates, and the group will be compared to the remaining groups G_n, $n \neq 1$. Consider the two objects c and j from Table 1. To determine if object c is preferable over object j in terms of *Rent*, then we would have to compute the probability of $Pr(c < j)$. Since c is an exact value, we can instead represent c as a continuous range $[c_{lb} : c_{ub}]$, where $c_{lb} = c_{ub}$ in order to conform it to the relations defined in Definition 5. The relationship between j and c can be seen as *contain-inverse*, thus we can use (7) to compute $Pr(c < j)$ as follows:

$$Pr(c < j) = Pr\{c_{ub} \leq j \leq j_{ub}\} + \frac{1}{2}(Pr\{c_{lb} \leq j \leq c_{ub}\}) \tag{9}$$

As $c_{lb} = c_{ub}$, then the probability that j has any exact value, for instance, c_{lb}, is 0. Thus, (9) can be simplified as follows:

$$Pr(c < j) = Pr\{c_{ub} \leq j \leq j_{ub}\} \tag{10}$$

By using the values in Table 1, then according to (10) we would get $Pr(c < j) = 0.8894$. This means that we are calculating the probability of c dominating j as the probability when j lies in its continuous range of $[c_{ub} : j_{ub}]$, that is $Pr(c_{ub} \leq j \leq j_{ub})$. However we would like to compute the probability of c dominating j as the probability when j strictly lies in its continuous range of $(c_{ub} : j_{ub}]$. Hence, in order to compute $Pr(c_{ub} < j \leq j_{ub})$ we modify (10) as follows:

$$Pr(c < j) = \lim_{\varepsilon \to 0^+} Pr\{c_{ub} + \varepsilon \leq j \leq j_{ub}\} \tag{11}$$

where $\lim_{\varepsilon \to 0^+}$ means that the limit is assumed as ε decreases to 0 and that ε represents a correction value that is as small as possible around c. As a result, the new $Pr(c < j)$ according to (11) when $\varepsilon = 0.5$ is 0.8890.

Therefore, to accommodate the dominating probability of any two objects with a continuous range $[v_{lb} : v_{ub}]$ and an exact value w, the relations defined in Definition 5 can be modified as follows:

Definition 8 (Range-exact value relations).
Disjoint: If $w \geq v_{ub}$

$$Pr(v < w) = 1 \tag{12}$$
$$Pr(w < v) = 0 \tag{13}$$

Disjoint-inverse: If $w \leq v_{lb}$

$$Pr(v < w) = 0 \tag{14}$$
$$Pr(w < v) = 1 \tag{15}$$

Contain: If $v_{lb} \leq w \leq v_{ub}$

$$Pr(v < w) = Pr\{v_{lb} \leq v \leq w - \varepsilon\} \tag{16}$$
$$Pr(w < v) = Pr\{w + \varepsilon \leq v \leq v_{ub}\} \tag{17}$$

Therefore, in the final step, the dominance tests are performed in a pairwise fashion between two local skyline candidates from different groups. This step employs probability calculations according to the relations defined in Definition 5 and Definition 8. If the computed probability for an object is below than the threshold value τ, then it is no longer needed to consider the object in any further computations. In doing the pruning process, it helps to reduce the unnecessary probability computations. And thus, all objects that manage to survive the final filtering process are considered as skyline objects according to Definition 7. Following from the running example, the final pruning process is applied on the surviving skyline candidates from each group. Skyline candidates from G_1 are now considered as the initial global skyline candidates. Thus, between the initial global skyline candidate c and local skyline candidate m in G_2, since $\forall i \in (3 \leq i \leq 4)$, $c.D_i = m.D_i$ and $c.D_2 < m.D_2$, therefore we can conclude that $c \prec m$ in $\mathbb{C}$ and c still is a global skyline candidate. And since $m_{[ub]} < c$ in $\mathbb{U}(D_1)$, then we can conclude that overall, c and m are incomparable. Similarly for objects k and m where they are both incomparable as k manages to not be dominated by m with a probability $Pr(k.D_1 < m.D_1) = 0.71$ (following (17)) that is at least τ in $\mathbb{UC}$. Between objects m and l however, m definitely dominates l in both $\mathbb{C}$ and $\mathbb{UC}$ and thus l is eliminated from the set of global skyline candidates and acknowledging m as a global skyline candidate. The same method is applied on the remaining local skyline candidates in G_3 and G_4, which as a result, makes n the only global skyline candidate that remain and thus *SkyQUD* will return n as the final skyline object. The general outline of the *SkyQUD* algorithm is presented in Algorithm 1.

Assume there is an algorithm that apply the same skyline probability computation as in *SkyQUD* but without having to partition the dataset into distinct groups. Then, the complexity of the algorithm is of the order $\mathcal{O}(nm)$, where n is

Algorithm 1. *SkyQUD*

```
Input: a $d$-dimensional dataset $S$ with $\mathbb{U}(D_i)$, $1 \leq i \leq d$, threshold $\tau$
Output: a set of objects, $Sky$, which is the skyline with probability $\geq \tau$
Initialise $Sky$: Skyline, $SkyC$: SkylineCandidates, and $G$: DistinctiveGroup
group $S$ into distinctive groups according to the existence of $\mathbb{U}(D_i)$
for each group $G_i \in G, (1 \leq i \leq \varrho)$ do
    if $G_i$ consists of objects with exact values then
        apply dominance test in Def. 1 on $G_i$
        if object $o \in G_i$ is not dominated then
            insert $o$ into $G_i.SkyC$
        end if
    else if $G_i$ consists of objects with continuous ranges and exact values then
        apply dominance test in Def. 6 on $G_i$
        if object $o \in G_i$ is not dominated then
            insert $o$ into $G_i.SkyC$
        end if
    end if
end for
Initialise $GSkyC = G_1.SkyC$
for each skyline candidate $sc_m \in G_i.SkyC, (2 \leq i \leq \varrho)$ do
    for each skyline candidate $sc_n \in GSkyC$ do
        apply dominance test in Def. 6 between $sc_m$ and $sc_n$
        if skyline candidate $sc_m$ is not dominated then
            insert $sc_m$ into $GSkyC$
        end if
    end for
end for
insert $GSkyC$ into $Sky$
return $Sky$
```

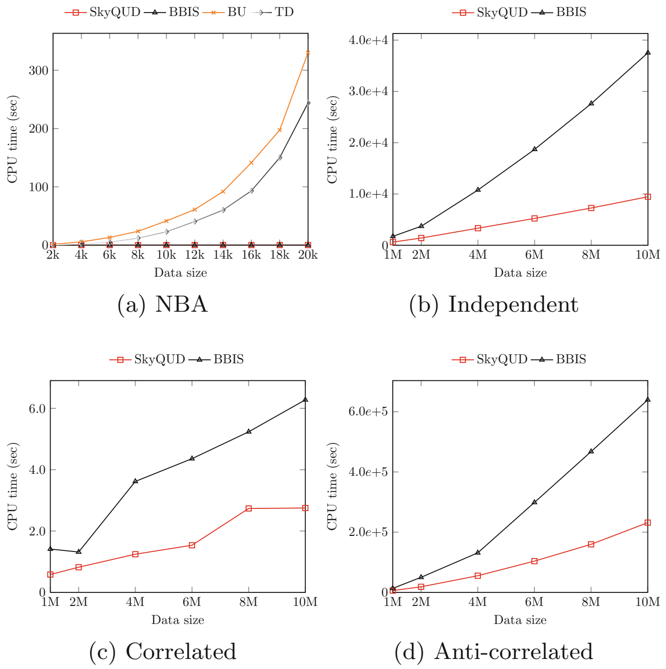
the total objects in the dataset and m is the total skyline candidates. Therefore, it can be assumed that the complexity of *SkyQUD* is of the order $\mathcal{O}(n_z m_z)$, such that $(1 \leq z \leq \varrho)$, where $n_z < n$ is the total objects in group z and $m_z < m$ is the total skyline candidates in group z.

4 Related Work

The evolution of skylines in the context of databases can be seen from the first work in [4]. Inspired by the work in [4,16] then proposed two algorithms to retrieve skyline objects. Subsequently, [10] introduced an online skyline algorithm based on the nearest neighbour search using R*-tree. Then, [7] introduced presorting into skyline computation algorithm to build a more effective algorithm. Following [7,8] have further improved it. To overcome the problem in [10,13] proposed an algorithm that is based on the sorted R-tree.

Moving in a different direction, [14] first pioneered the concept of probabilistic skyline on uncertain data, in which each object is represented by multiple instances and is part of the skyline answer with a certain probability. They proposed two algorithms, namely: *bottomUp* (*BU*) and *topDown* (*TD*), to answer probabilistic skyline queries (p-skyline) on uncertain data. Later, [17] followed with their work by defining the concept of skyline probability for uncertain data with maybe confidence, [3] introduced τ-skyline query which applies Gaussian Mixture Models on the probability density function in order to answer skyline query on uncertain data, while [1] addressed on the issue of computing exact skyline probabilities for all objects at the instance level with no threshold value.

Influenced by [9,14] then introduced another interesting concept of probabilistic skyline which focuses on uncertainty in data in continuous domains where an object is represented as a continuous range in one of its dimension, which is associated with a probability density function capturing the likelihood of possible values. Later, [11] followed with their work where the value of an attribute for uncertain objects can be represented as exact values or intervals that conform to a probability distribution. Therefore, they have proposed a progressive algorithm, named *Branch-and-Bound Interval Skyline* (*BBIS*) which is modified from [13] to process the interval skyline query with an optimal costs of I/O. The algorithm employs R*-tree to index the interval objects, and this ensures that the algorithm performs only a single access to all nodes that may contain skyline objects.



(a) NBA

(b) Independent

(c) Correlated

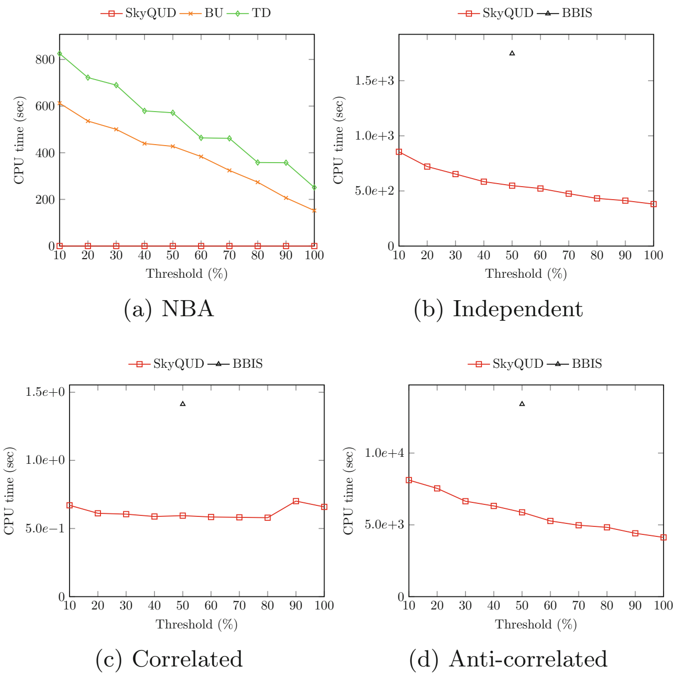
(d) Anti-correlated

Fig. 3. Effect of n on processing time.

5 Empirical Study

In this section, we perform extensive experiments to verify the effectiveness of *SkyQUD* algorithm and compared it to *BBIS* [11], *BU* [14], and *TD* [14]. All experiments were conducted on a PC with Intel Core i5-3470 3.20 GHz processor and 7.8 GB main memory running Ubuntu Linux operating system.

Following [9], for synthetic datasets we generate a d-dimensional dataset of n objects, where d is varied from 3 to 20 and n is varied from 1M to 10M where each dimension represents a uniform random variable from 1 to 10,000. We have set the first dimension to be the dimension that will represent the concept of *uncertain dimension* (i.e. $\mathbb{U}(D_1)$), where the distributions (δ) between exact values and continuous ranges in $\mathbb{U}(D_1)$ by default is set to be equally distributed. On the other hand, the NBA statistics contains records of 21,961 NBA players from year 1946 to 2009. Each record has 16 dimensions that represent various statistics associated with basketball games. However, the NBA statistics is initially represented in exact values that are certain and complete in nature, therefore, following [9], we have to explicitly add another dimension in order to introduce the concept of *uncertain dimension* to the dataset. The *uncertain dimension* is generated following the same settings used in generating synthetic datasets. Note that for *BU* and *TD* algorithms, we sample a number of random points from a continuous range to represent instances for each object with continuous



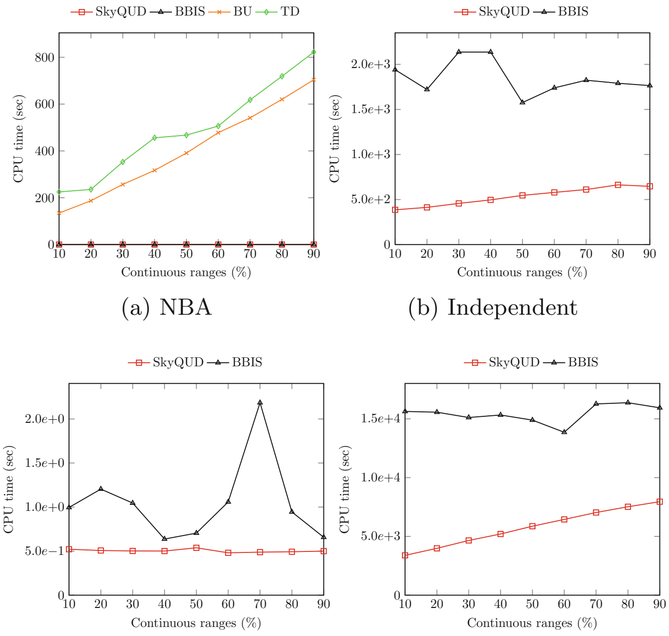
(a) NBA

(b) Independent

(c) Correlated

(d) Anti-correlated

Fig. 4. Effect of τ on processing time.



(a) NBA

(b) Independent

(c) Correlated

(d) Anti-correlated

Fig. 5. Effect of δ on processing time.

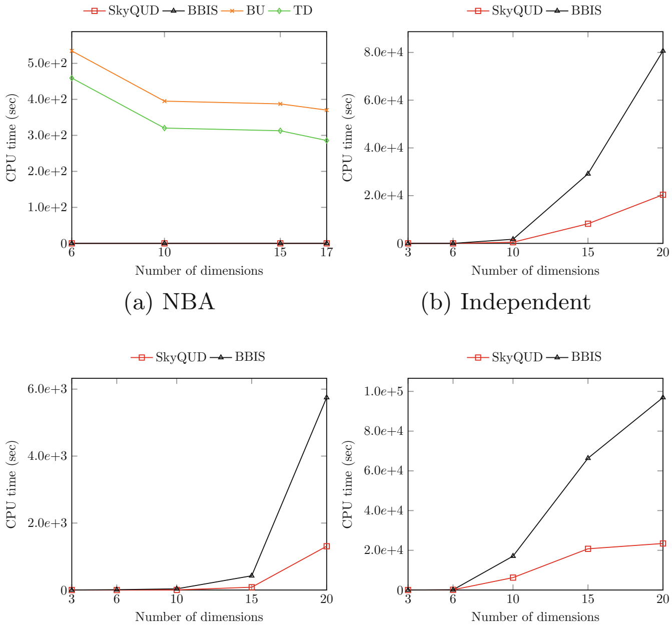
range in order to simulate the datasets used within these two algorithms. We use the same parameters described in [14] to generate the instances.

Scalability. Figure 3 shows the scalability of our algorithm in terms of processing time when increasing the data size n (objects) from 2k to 20k for real dataset and from 1M to 10M for synthetic datasets. However, *BU* and *TD* fail to terminate on synthetic datasets with $n > 1\text{M}$ (Fig. 3b, c, and d) while their performance is clearly worse than the other algorithms on NBA dataset (Fig. 3a) due to the tremendous amount of instances ($\frac{n\delta u}{2}$) to be processed as compared to the number of objects (n) processed in *SkyQUD* and *BBIS*. *SkyQUD* managed to outperformed *BBIS* on all datasets due to the implementation of an R*-tree index structure in *BBIS* to index the datasets.

Effect of Threshold. Figure 4 presents the effect of increasing the threshold value τ in terms of processing time when τ is varied from 0.1 to 1.0 for both the real and synthetic datasets. *SkyQUD* as well as *BU* and *TD* (Fig. 4a) exhibit an increment of speed in their performance together with the increase of τ. As *BBIS* does not implement any probability computations, therefore the parameter threshold τ does not affect the algorithm. However, in the interest of thoroughly evaluating the performance of *SkyQUD*, we have compared the processing time of *BBIS* to *SkyQUD* when the parameter threshold τ in *SkyQUD* is set to 50%. This is due to the fact that every continuous range is treated as its median value in *BBIS*. This median value normally reflects the continuous range when it is at its average case scenario. The results clearly indicate that *SkyQUD* performs better than *BBIS* on all datasets.

Effect of Data Distribution. Figure 5 illustrates the behavior of *SkyQUD*, *BBIS*, *BU*, and *TD* in terms of processing time when distributions of exact values and continuous ranges in the *uncertain dimension* are varied. We increase the distributions of data from $\delta_{\mathbb{U}} = 10\%$ for data with continuous ranges and $\delta_{\mathbb{C}} = 90\%$ for data with exact values to $\delta_{\mathbb{U}} = 90\%$ and $\delta_{\mathbb{C}} = 10\%$. From the experiments executed, it can be seen that *BBIS* is not largely affected by the distributions of continuous ranges and exact values in the *uncertain dimensions*. This is due to the implementation of *BBIS* that does not compute any object probabilities as they treat every object with continuous range as an exact value, which is obtained by the median value of the continuous range. Nevertheless, *SkyQUD* managed to outperformed *BBIS* mainly due to the poor performance of R*-tree and the implementation of objects partitioning in *SkyQUD*. On the other hand, the performance of *BU* and *TD* is largely affected as the number of instances generated ($\frac{n\delta u}{2}$) is dependent on the distributions of data with continuous ranges, that is $\delta = \delta_{\mathbb{U}}$. Thus, the higher $\delta_{\mathbb{U}}$, the slower the speed of *BU* and *TD*.

Effect of Dimensionality. Figure 6 shows the processing time of the *SkyQUD* and *BBIS* algorithms as the number of dimensions is varied from 3 to 20 on 1M



(a) NBA (b) Independent

(c) Correlated (d) Anti-correlated

Fig. 6. Effect of d on processing time.

objects. Both algorithms follow similar trends where the processing time for all algorithms increases with the increase of the number of dimensions. Conversely, the *BBIS* algorithm outperforms the *SkyQUD* algorithm in terms of speed when the number of dimensions is less than 5, yet its performance decreases rapidly when the number of dimensions is increased from 6 to 20. The performance of R*-tree is known to rapidly deteriorates when handling data with higher dimensions due to the rapid increase of overlapping regions in the directory [2,11,13]. Meanwhile, Fig. 6a indicates that *BU* and *TD* perform faster as the dimensionality increases due to the decreases in the average number of possible dominating objects for each object since the dataset becomes sparser when the dimensionality increases [14].

6 Conclusion

In this paper, we define the concept of *uncertain dimensions* as dimensions that contain continuous ranges and/or exact values. We define seven types of relations to determine the dominance relation between any two objects with continuous ranges. In contrast, to accommodate the dominance relation between any two objects with a continuous range and exact value, we modify the former relations by employing a correction value ϵ in each of the probability computations. We define our skyline object on *uncertain dimensions* to be objects that are not dominated with a dominance probability that is at least τ. We propose *SkyQUD* algorithm for processing skyline queries on high dimensional data with *uncertain dimensions*. Our performance study shows that the *SkyQUD* algorithm is efficient and scalable for analysing skyline queries on high dimensional data with *uncertain dimensions*.

Acknowledgements. This research was supported by Ministry of Science, Technology, and Innovation under the Fundamental Research Grant Scheme (Grant no. 08-01-16-1853FR). All opinions, findings, conclusions and recommendations in this paper are those of the authors and do not necessarily reflect the views of the funding agencies. We thank the anonymous reviewers for their comments.

References

1. Atallah, M., Qi, Y.: Computing all skyline probabilities for uncertain data. In: Proceedings of the ACM SIGMOD-SIGACT-SIGART Symposium of the Principles of Database Systems (PODS), pp. 279–287 (2009)
2. Berchtold, S., Keim, D.A., Kriegel, H.P.: The X-tree: an index structure for high-dimensional data. In: Proceedings of the 22nd International Conference on Very Large Data Bases (VLDB), pp. 28–39 (1996)
3. Böhm, C., Fiedler, F., Oswald, A., Plant, C., Wackersreuther, B.: Probabilistic skyline queries. In: Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM), pp. 651–660 (2009)
4. Börzsönyi, S., Kossmann, D., Stocker, K.: The skyline operator. In: Proceedings of the 17th International Conference on Data Engineering (ICDE), pp. 421–430 (2001)

5. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: On high dimensional skylines. In: Ioannidis, Y., et al. (eds.) EDBT 2006. LNCS, vol. 3896, pp. 478–495. Springer, Heidelberg (2006). https://doi.org/10.1007/11687238_30
6. Chan, C.-Y., Jagadish, H.V., Tan, K.-L., Tung, A.K.H., Zhang, Z.: Finding k-dominant skylines in high dimensional space. In: Proceedings of International Conference on Management of Data (SIGMOD), pp. 503–514 (2006)
7. Chomicki, J., Godfrey, P., Gryz, J., Liang, D.: Skyline with presorting. In: Proceedings of International Conference on Data Engineering (ICDE), pp. 717–816 (2003)
8. Godfrey, P., Shipley, R., Gryz, J.: Maximal vector computation in large data sets. In: Proceedings of International Conference on Very Large Data Bases (VLDB), pp. 229–240 (2005)
9. Khalefa, M.E., Mokbel, M.F., Levandoski, J.J.: Skyline query processing for uncertain data. In: Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), pp. 1293–1296 (2010)
10. Kossmann, D., Ramsak, F., Rost, S.: Shooting stars in the sky: an online algorithm for skyline queries. In: Proceedings of International Conference on Very Large Data Bases (VLDB), pp. 275–286 (2002)
11. Li, X., Wang, Y., Li, X., Wang, G.: Skyline query processing on interval uncertain data. In: IEEE 15th International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing Workshops, pp. 87–92 (2012)
12. Mokbel, M.F., Levandoski, J.J.: Toward context and preference-aware location-based services. In: Proceedings of the International Workshop on Data Engineering for Wireless and Mobile Access, pp. 25–35 (2009)
13. Papadias, D., Tao, Y., Fu, G., Seeger, B.: Progressive skyline computation in database systems. ACM Trans. Database Syst. **30**(1), 41–82 (2005)
14. Pei, J., Jiang, B., Lin, X., Yuan, Y.: Probabilistic skylines on uncertain data. In: Proceedings of International Conference on Very Large Data Bases (VLDB), pp. 15–26 (2007)
15. Ross, S.M.: Introduction to Probability Models, 8th edn. American Press, San Diego (2003)
16. Tan, K.L., Eng, P.K., Ooi, B.C.: Efficient progressive skyline computation. In: Proceedings of International Conference on Very Large Data Bases (VLDB), pp. 301–310 (2001)
17. Yong, H., Kim, J.-H., Hwang. S.-W.: Skyline ranking for uncertain data with maybe confidence. In: Proceedings of the 2008 IEEE 24th International Conference on Data Engineering Workshop (ICDEW), pp. 572–579 (2008)

Cognitive Computing

Semi-automatic Quasi-morphological Word Segmentation for Neural Machine Translation

Jānis Zuters(✉), Gus Strazds, and Kārlis Immers

University of Latvia, Raina blvd. 19, Riga 1586, Latvia
janis.zuters@lu.lv, gstrazds@gmail.com,
karlis.immers@gmail.com

Abstract. This paper proposes the Prefix-Root-Postfix-Encoding (PRPE) algorithm, which performs close-to-morphological segmentation of words as part of text pre-processing in machine translation. PRPE is a cross-language algorithm requiring only minor tweaking to adapt it for any particular language, a property which makes it potentially useful for morphologically rich languages with no morphological analysers available. As a key part of the proposed algorithm we introduce the 'Root alignment' principle to extract potential sub-words from a corpus, as well as a special technique for constructing words from potential sub-words. We conducted experiments with two different neural machine translation systems, training them on parallel corpora for English-Latvian and Latvian-English translation. Evaluation of translation quality showed improvements in BLEU scores when the data were pre-processed using the proposed algorithm, compared to a couple of baseline word segmentation algorithms. Although we were able to demonstrate improvements in both translation directions and for both NMT systems, they were relatively minor, and our experiments show that machine translation with inflected languages remains challenging, especially with translation direction towards a highly inflected language.

Keywords: Neural machine translation
Processing morphologically rich languages · Word segmentation

1 Introduction

In recent years neural machine translation (NMT) has indisputably become the default approach for machine translation. Still, the quality of translation differs widely depending on the language pairs involved – for morphologically rich languages, especially those with relatively small amounts of available parallel training data, training an NMT system remains challenging due to data sparseness [1].

To overcome data sparsity due to inflectedness of a language, it is common to apply various forms of data pre-processing, and one of the most commonly used techniques is splitting words into segments (or sub-words) in order to decrease the amount of unique input tokens. This reduces data sparseness to the extent that a large number of lexicographically unique word tokens (in morphologically rich languages, these include the many inflected forms of each individual word) can be represented as combinations built

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 289–301, 2018.
https://doi.org/10.1007/978-3-319-97571-9_23

up from a much smaller vocabulary of sub-word tokens. This is important because of the main paradigm of NMT – sequence to sequence transduction of text units (characters, or words, or sub-words) that are seen and processed by the system as indivisible tokens. A good segmentation into sub-word tokens would have the property that specific word forms which were not present in the training data at all can nevertheless be represented as a sequence of tokens from the sub-word vocabulary, and, ideally, the neural network could learn to generate correct (but possibly not previously encountered) output sequences for previously unseen specific input sequences (e.g. producing word forms that are correctly inflected even though they might not have been present in the training datasets).

This article focuses on word segmentation implicitly based on sub-word statistics (Prefix-Root-Postfix-Encoding algorithm, PRPE). The output text resembles morphologically segmented text, but without making any claims to being a linguistically well-motivated morphological splitting. Thus, the output of the proposed segmentation method was not compared against a reference segmentation (as is done, for example, in [2]). Producing reference segmentations for a large corpus of text requires considerable effort (and, usually, some amount of non-trivial linguistic theory concerning the morphological structure of the specific language begin analysed). Instead, experiments were conducted to directly test whether PRPE segmentation improves translation quality relative to a couple of baseline segmentation schemes. Unlike language specific morphological segmenters, PRPE is almost language independent, requiring relatively little work (a handful of new or modified lines of code and some parameter tuning) to adapt it to a new language.

2 Related Work

This paper focuses on a particular approach of text pre-processing for NMT to overcome inflectedness of languages and the resultant problem of sparsity of specific word forms in training corpora – segmentation of text into sub-word units. This section gives a brief overview of some commonly used sub-word segmentation algorithms.

2.1 Byte Pair Encoding Based Segmentation Algorithm

Byte pair encoding based segmentation algorithm (BPE), proposed in [3], utilizes the principle of iteratively finding the most frequent character sequences of the text to become potential segments (see a segmentation example in Table 1).

The algorithm consists of two phases: (a) the learning phase, in which the vocabulary of merge operations is obtained, (b) the apply phase, in which a specific text is segmented using the vocabulary.

The learning phase starts with all words in the text represented as sequences of characters. Then, through an iterative process, the most frequent pairs of neighbouring symbols (initially, characters) are merged together and these pairs (or 'merge operations') are written to a special vocabulary. At each iteration, (a) the chosen merge operation is added to the vocabulary, (b) the merge operation is applied to the text. The process is continued until a predefined number of merge operations is reached.

Table 1. A segmentation example with BPE

Language	Segmented text
English	you need to know exactly what you want to im–mor–tal–ise during the photo session, and be able to tell the photo–grap–her about it
Latvian	ir jāsaprot, ko tieši tu vē–lies ie–mūž–ināt foto–sesijas laikā un jāpa–stāsta par to fotogrāf–am

The apply phase transforms input text into segmented text according to the vocabulary of merge operations.

BPE provides control over the effective size of the vocabulary for translation, since the vocabulary of unique tokens after applying BPE is less than or equal to the number of unique characters in the original input text plus the number of merge operations. A bounded vocabulary is essential for typical approaches to NMT, and since the introduction of the BPE algorithm adapted for this purpose in [3], it has become something of a standard practice to pre-process input text for NMT by segmenting it with BPE.

2.2 Morphology-Driven Splitting

One of ideas for word segmentation for NMT is trying to separate roots from affixes (especially suffixes in morphologically rich languages), in the hope that doing so will preserve more semantic information (words with common roots would also have the same segments).

In [1] a language-specific morphological splitting approach is described (see a segmentation example in Table 2). To avoid over-segmentation of the text, morphological splitting is performed in a limited manner, i.e., not all affixes are separated (too many segments in a sequence reduces the quality of NMT).

Table 2. A segmentation example with morphology-driven splitting proposed by [1] (postprocessed with BPE to support open vocabulary)

Language	Segmented text
English	you need to know exact–ly what you want to im–mor–tal–ise during the photo session, and be able to tell the photo–grap–her about it
Latvian	ir jā–saprot, ko tieš–i tu vēl–ies ie–mūž–inā–t foto–sesij–as laik–ā un jā–pastāst–a par to foto–grāf–am

For translation between English and Latvian, morphology-driven splitting was found to give a small improvement on translation quality (0.5–0.7 BLEU points, [4]) relative to BPE. The small improvement might be explained by a relatively small out-of-vocabulary rate given the training data used (especially in English).

Morphology-driven splitting is typically carried out using language-specific morphological analyzers. Building such analyzers for inflective and agglutinative languages is more complicated than for English (see [5]). For example, for many languages morphological analysis must deal with a considerable amount of ambiguity, and therefore various disambiguation models are used [6, 7]. As morphological analyzers are typically language specific, it takes a lot of effort to build such a tool for any given language (e.g. creating morphologically annotated corpora, developing language specific routines).

2.3 Morphological Segmentation Using Morfessor

Morfessor is a toolkit for morphological segmentation of agglutinative and inflected languages that's used for morphological analysis, speech recognition and machine translation (see a segmentation example in Table 3).

Table 3. A segmentation example with Morfessor segmentation proposed by [8] (post-processed with BPE to support open vocabulary)

Language	Segmented text
English	you need to know exact–ly what you want to im–mortal–ise dur–ing the photo session, and be able to tell the photograph–er about it
Latvian	ir jā–saprot, ko tieš–i tu vēl–ies ie–mūž–inā–t foto–sesij–as laik–ā un jā–pastāst–a par to foto–grāf–am

Morfessor produces a language-independent segmentation model. It works by attempting to split each word into two sub-word units in all possible ways. For each set of two sub-word units, the number of occurrences in the training data is counted. The combination with the highest score is recursively processed again, until the undivided string returns a higher score than any split. The least amount of highest rated sub-word units that are required to construct the original word are added to segmentation model [8].

Morfessor's segmentation process uses the Viterbi algorithm to split text into sub-word units given the learned segmentation model. The Viterbi algorithm is a hidden Markov model decoding algorithm for finding the most probable sequence of hidden states that could have caused a given sequence of observations [9]. In the case of word segmentation, the Viterbi algorithm is used to find the most probable sequence of sub-word units that produce the given word.

Training data segmentation using Morfessor is reported to give small improvements on statistical machine translation quality, especially for inflected languages (up to 0.41 BLEU points, [10, 11]). In our initial experiments, however, we didn't achieve competitive results with text segmented using Morfessor. That's why it is not included in our resulting reports.

3 PRPE Segmentation Algorithm

3.1 General Description

This section describes the basic principles of the proposed Prefix-Root-Postfix-Encoding (PRPE) algorithm[1] (see a segmentation example in Table 4). The main motivation for the algorithm is the belief that splitting away roots from words would produce more meaningful parallel sequences for machine translation (as with morphology-driven splitting, see Sect. 2.2), thus increasing the quality of machine translation. But the goal of PRPE is to obtain such a segmentation based primarily on the statistics of the training data, using a bare minimum of language specific knowledge (contrast this with a language-specific morphological analyser, which would be hand-crafted using a large number of language-specific rules based on a linguistically motivated analysis of the morphological processes at work in a given language).

Table 4. A segmentation example with PRPE (post-processed with BPE to support open vocabulary). Linguistically, morphological splitting is similar in Latvian and English. The two main differences for Latvian: (1) substantially more inflectedness = many more systematically varying word endings; and (2) word roots almost always end with a consonant.

Language	Segmented text
English	you need to know exactly what you want to im–mortal–ise dur–ing the photo session, and be able to tell the photo–graph–er about it
Latvian	ir jāsaprot, ko tieš–i tu vēl–ies ie–mūž–ināt foto–sesij–as laik–ā un jāpa–stāst–a par to foto–grāf–am

The basic principle underlying PRPE comes from the BPE algorithm – to learn the most frequent character sequences and then use them to segment words in a text. The main idea added is to take the most frequent left and right substrings of words instead of any character sequences, regarding left substrings as potential prefixes and roots, but right substrings as potential postfixes. Then these potential building blocks (prefixes, roots, postfixes) are combined together in a special way to constitute words – thus performing segmentation. As a result, a close-to-morphological segmentation is obtained. For better results, the PRPE algorithm should be complemented with a small language specific part. Instead of complicated probability computations, in PRPE we use substring frequencies and lists of substrings specifically ranked according to frequencies.

PRPE has two phases:

- The **learning phase**, in which ranked lists of main building blocks (potential prefixes, roots and postfixes) are obtained;
- The **application phase**, in which segmentation is performed using obtained building blocks.

[1] Source code available at: https://github.com/zuters/prpe.

From the algorithmic perspective, PRPE contributes two main ideas:

- The ‘Root alignment’ principle to extract potential roots and other sub-words in the learning phase;
- A special technique to construct words from obtained potential sub-words thus accomplishing word segmentation.

3.2 Obtaining Potential Segments

The main goal of the learning phase of PRPE is to obtain lists of potential prefixes, roots and postfixes (suffixes and endings) from a single-language corpus.

<table>
<tr><th>un</th><th>believ</th><th>abl</th><th>es</th></tr>
<tr><td rowspan="2">prefix</td><td rowspan="2">root</td><td>suffix</td><td>ending</td></tr>
<tr><td colspan="2">postfix</td></tr>
</table>

Fig. 1. Illustration of the building blocks used in PRPE for the word “unbelievables”

The key idea of the algorithm is the ‘Root alignment’ principle (see illustrations in Figs. 1 and 2, and example of implementation in Fig. 3):

- left substrings of words are considered potential roots;
- aligning potential roots with the middle parts of words allows extracting potential prefixes and postfixes.

<table>
<tr><th>u</th><th>n</th><th>b</th><th>e</th><th>l</th><th>i</th><th>e</th><th>v</th><th>a</th><th>b</th><th>l</th><th>e</th><th>s</th></tr>
<tr><td colspan="2">prefix</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="4">potential root</td><td></td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="5">potential root</td><td></td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="6">potential root</td><td></td><td></td><td></td><td></td><td></td></tr>
<tr><td></td><td></td><td colspan="7">potential root</td><td></td><td></td><td></td><td></td></tr>
</table>

Fig. 2. The illustration of the ‘Root alignment’ principle in word “unbelievables”: potential roots aligned with the middle part of the word to collect statistics for prefix “un”

Obtaining potential segments is carried out in four steps:

1. Collecting frequency statistics of left and right substrings of words. For instance, among the most frequent left substrings in English we can found “the”, “ther”, “re”, “commis”, but among the most popular right substrings – “s”, “es”, “tion”, “ation”.
2. Extracting potential prefixes from left substrings through aligning other left substrings as potential roots with the middle part of word:
   a. obtain prefix statistics,
   b. select the most frequent prefixes to become potential prefixes in segmentation.

3. Extracting potential postfixes from right substrings through aligning other left substrings as potential roots with the middle part of word:
   a. obtain postfix statistics (in a similar way as for prefixes),
   b. select endings from postfixes according predefined rules to become potential endings in segmentation;
   c. extract and select the most frequent suffixes from postfixes by splitting away collected endings – to become potential suffixes in segmentation.
4. Extracting potential roots from left substrings through aligning them with the middle part of word considering already collected prefixes and postfixes. Here longer roots are also assigned bigger weight coefficients to better compete with smaller roots in the segmentation phase.

All the obtained lists of potential sub-words are ranked, and the predefined hyper-parameters determine how many of the respective sub-words will become final building blocks. Ranking numbers (1, 2, 3, etc.) will be then used to calculate the best segmentation.

As postfixes are split into suffixes and endings (which is not so important for English, but matters for morphologically rich languages), the output of the learning phase consists of four ranked lists: prefixes, roots, suffixes and endings.

```
module extract_potential_prefixes (vocab, leftstat):
  vocab - list of all words found in the text corpus
  leftstat - statistics of frequencies of left substrings
  prefstat - prefix statistics to be calculated
  for each word w in the vocabulary vocab:
    for each left substring p in w: # a potential prefix
      if p is a valid prefix according to a hardcoded control:
          # a potential root in the middle of w:
        for each substring r in w just after p:
          if r is a valid root according to a hardcoded control
          and r is found in leftstat:
            prefstat[p] = prefstat[p] + leftstat[r]
  return prefstat
```

Fig. 3. Prefix extraction module to algorithmically illustrate the ‘Root alignment’ principle: trying to locate potential roots (frequent left substrings) in the middle of a word to extract potential prefixes. Extracting postfixes is designed using the same approach.

3.3 Segmenting Words Using Obtained Potential Segments

The segmentation phase uses ranked lists (prefixes, roots, suffixes and endings) to segment words. Ranking numbers are used to calculate the best segmentation candidate.

Segmenting a word is carried out in the following way:

1. all possible segmentations for the word are obtained;
2. the highest ranked candidate segmentation wins.

Collecting All Possible Segmentations. Four ranked lists of potential segments available (P: prefixes, R: roots, S: suffixes and E: endings) for segmentation. Each candidate segmentation is built in the following form:

$$([p]\,[p]\,r\,[s]\,[e])\;+, \tag{1}$$

where $p \in$ P, r $\in$ R, s $\in$ S, e $\in$ E.

This means that one segmentation is one or more ‘root blocks’ (as root is the only mandatory block in the big block). We search for two prefixes because the two prefix case is quite common in Latvian (an example from English would be “non-re-active”).

Example of segmentation candidates for word “unbelieve” (‘/’ marks boundary of two candidate ‘root blocks’):

- un–bel–ieve
- un–bel–i / eve
- un–believ–e
- un–believe

Calculating the Best Segmentation. The best segmentation is the highest ranked segmentation from those with the smallest number of ‘root blocks’, and the rank of the segment is the sum of ranks of individual blocks. In the example above the segmentation #2 is of two ‘root blocks’, i.e., out of competition.

3.4 Additional Heuristics

Several addition heuristics were used to tune the algorithm for better results.

Most Frequent Words Unsegmented. To reduce the final number of segments, a predefined number of the most frequent words stay unsegmented (see ‘leave-out rate’ in the results).

Optimization of the Segmentation. To reduce the final number of segments, several heuristics are used to join back some segments, e.g.:

- prefixes not split away,
- suffixes not split away between roots.

No Segmentation Candidates. If there are no segmentations candidates (i.e., a word cannot be built using available blocks), only the best postfix is split away.

Uppercase Marking. A word starting with uppercase and the rest symbols in lowercase converted to lowercase and a special uppercase marker is inserted before it.

3.5 Adapting the Algorithm to a Particular Language

As the algorithm is not fully language-independent, some minor adaptation should be carried out for a particular language:

1. add a small amount of language-specific source code (candidate word parts are additionally screened by a small number of hand-coded routines/rules);
2. tune hyperparameters (e.g., how many prefixes should be selected as potential prefixes, minimum length of prefixes).

According to the experiments, adapting to a particular language noticeably increases the segmentation quality.

4 Experiments and Results

The main idea for the experiments was to show that pre-processing corpora with PRPE yields better machine translation results, relative to baseline segmentation schemes.

For our experiments, we used the English-Latvian dataset provided in the WMT 2017[2] shared task in news translation. The approximate size of each of the parallel corpora – 1.6M sentences. We use as a starting point the data as pre-processed (filtered, normalised, tokenised) by the authors of [12] for their experiments.

We obtained sub-word-segmented versions of both the English and Latvian texts using various configuration of PRPE, as well as several baseline segmentation algorithms:

1. BPE [3][3];
2. Tilde's Morphologically segmented version of the same dataset, also provided to us by the authors of [1, 12];
3. the same dataset segmented using Morfessor [8, 13].

All the non-BPE segmentations were also post-processed using BPE to better support open-vocabulary translation (by ensuring full coverage of the word vocabulary in the training data, since that is not an explicit goal/guarantee of the alternative segmentation schemes). In all cases, both languages were segmented similarly, using the same algorithm with one set of configuration parameters per experiment.

To evaluate the impact of PRPE on machine translation, we then used these various sub-word-segmented parallel corpora to train English-to-Latvian (en-lv) and Latvian-to-English (lv-en) translation models using two architecturally quite different NMT systems:

1. Nematus [14][4] is a framework for NMT based on what has become essentially the standard reference architecture for learning sequence to sequence translation tasks: encoder-decoder using recurrent neural networks, with an attention mechanism that

[2] http://www.statmt.org/wmt17/translation-task.html.

[3] https://github.com/rsennrich/subword-nmt.

[4] https://github.com/EdinburghNLP/nematus.

gives the decoder access to richer information about the input sequence than what the encoder can encode into a fixed-length vector. For our primary baseline, we chose a relatively basic, straightforward configuration of the many options supported by Nematus: Hidden layer size = 1024, word embedding dimensions = 500, batch_size = 60, max length for input sequences = 80, no dropout, optimization using Adadelta, with early stopping after loss computed on a cross-validation dataset fails to improve for 10 × 10,000 batches. Default values were used for the depth of the recurrence transitions in the encoder and decoder (=1 and 2, respectively).

2. ConvS2S ("Convolutional Sequence to Sequence", [15])[5] is a newer architecture that has recently posted some new state-of-the results for NMT. Instead of recurrent neural networks, it uses convolutional networks for its encoder and decoder, a design choice which allows for greater parallelism when training the model (enabling significantly faster training). For our baseline configuration we simply used one of the default configurations included with the framework: "fconv_wmt_en_ro", a configuration originally used for an English->Romanian NMT model. It is a fairly deep model, with 20 layers in each of its encoder and decoder networks, word embedding dimensions = 512, hidden layer size = 512.

Training even a relatively small NMT model on one or two GPUs takes a minimum of several days, so resource and time constraints precluded our doing much in the way of search over the space of potential configuration and training hyperparameters for the NMT systems we used. But since our goal was not to find optimal configurations and maximize translation BLEU scores, but instead to test for incremental benefits from using our proposed sub-word segmentation scheme, we chose an initial set of NMT configuration and training parameters (yielding reasonably good baseline results), and then used them unchanged for all subsequent experiments. We did, however, try various settings of the internal parameters of the PRPE algorithm, and found that different settings yielded best results for Nematus vs. ConvS2S. This leads to the observation that PRPE configuration should be tuned in concert with other hyperparameters when training an NMT system. (This is completely analogous to selecting the number of merge operations for BPE.) In particular, the "leave-out rate" seems to be the most important tunable parameter for PRPE.

Previous results[6] have shown that the translation direction English-to-Latvian generally yields worse scores than Latvian-to-English, and in all cases our results were consistent with this finding. This could be explained by the supposition that translation towards a morphologically more rich language is a more challenging task. That's why we hoped to obtain improvements in this particular direction. Unfortunately, with Nematus, the best configuration of PRPE gave a minor (but not statistically significant[7]) improvement in BLEU score for lv-en (Latvian-to-English) translation, but in the

[5] https://github.com/facebookresearch/fairseq-py.

[6] http://www.statmt.org/wmt17/results.html.

[7] Statistical significance was estimated via bootstrap resampling using the script analysis/bootstrap-hypothesis-difference-significance.pl from the Moses MT system: https://github.com/moses-smt/mosesdecoder.

Table 5. Translation results with Nematus system using various segmentation techniques

	BPE (BLEU)	Tilde's morph (BLEU)	PRPE (leave-out rate = 5000)	
			BLEU	p-val vs BPE
en-lv	17.05	17.15	**17.16**	0.23
lv-en	18.66	18.67	**18.90**	0.13

Table 6. Translation results with ConvS2S system using various segmentation techniques

	BPE (BLEU)	Tilde's morph (BLEU)	PRPE (leave-out rate = 5000)	
			BLEU	p-val vs BPE
en-lv	20.30	21.26	**21.33**	0.00
lv-en	21.93	22.05	**22.61**	0.01

en-lv direction produced almost identical scores to the morphologically segmented baseline (see Table 5). With ConvS2S we observed statistically significant improvements in both directions (see Table 6).

Note that the baseline scores that we obtained using ConvS2S were 3-4 BLEU points higher than the corresponding scores obtained using Nematus with the same datasets. We conjecture that this might be to a large extent because we were using a relatively basic (shallow) configuration of Nematus, with less modeling capacity than the large and deep default configuration we chose for ConvS2S. To test this conjecture – and the possibility that deeper networks might be better able to make use of more sophisticated sub-word segmentation schemes (as suggested by the bigger boost from PRPE that we saw with ConvS2S vs. Nematus) – we ran a few additional experiments using a configuration for Nematus based on training scripts provided by Edinburgh University[8] [16], which make use of some Nematus features that allow for using deeper network configurations in its encoder and decoder [17]. Initial results (see Table 7) seem to confirm these conjectures, but, due to time constraints, a more systematic exploration will have to await future work.

Table 7. Translation results using deeper Nematus models

	BPE (BLEU)	PRPE (leave-out rate = 5000)	
		BLEU	p-val vs BPE
en-lv	19.13	**19.55**	0.06
lv-en	20.90	**21.46**	0.01

[8] http://data.statmt.org/wmt17_systems/training.

5 Conclusion

In this paper, we propose an algorithm for close-to-morphological word segmentation for machine translation without requiring the availability of language specific morphologically labelled data. Experimental results demonstrated that PRPE pre-processing of training data for NMT can yield small improvements in translation output, relative to pre-processing with baseline sub-word segmentation algorithms. But the results also show that machine translation with inflected languages remains a big challenge, especially with translation direction towards a highly inflected language.

The PRPE algorithm exploits the 'Root alignment' principle to extract potential sub-words, as well as a special technique to construct words from potential sub-words.

In addition, the experiments showed that fully splitting all affixes is counter-productive, in that it produces too long sequences of sub-words, and the translation quality grows worse. The best results were achieved with only compound splitting plus splitting postfixes from the end of a word, as well as leaving up to 5000 of the most frequently encountered words unsegmented.

Obtained improvements in translation quality from PRPE pre-processing were not particularly large, in some cases falling below a commonly used threshold for statistical significance, which might be a signal that the approach of autonomous (without using syntactic and semantic context) pre-processing to do sub-word segmentation might be near its limits for potential improvements.

Other already started experiments beyond the scope of this paper include running experiments training NMT models for other language pairs, as well as using parallel corpora to improve PRPE segmentation.

Acknowledgements. The research has been supported by the European Regional Development Fund within the research project "Neural Network Modelling for Inflected Natural Languages" No. 1.1.1.1/16/A/215, and the Faculty of Computing, University of Latvia.

References

1. Pinnis, M., Krišlauks, R., Deksne, D., Miks, T.: Neural machine translation for morphologically rich languages with improved sub-word units and synthetic data. In: Ekštein, K., Matoušek, V. (eds.) TSD 2017. LNCS (LNAI), vol. 10415, pp. 237–245. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-64206-2_27
2. Ruokolainen, T., Kohonen, O., Sirts, K., Grönroos, A., Kurimo, M., Virpioja, S.: A comparative study of minimally supervised morphological segmentation. Comput. Linguist. **42**(1), 91–120 (2016)
3. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016), Berlin, Germany (2016)
4. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: ACL 2002: 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)

5. Hajič, J.: Morphological tagging: data vs. dictionaries. In: Proceedings of the 1st North American Chapter of the Association for Computational Linguistics conference (NAACL 2000), pp. 94–101 (2000)
6. Paikens, P., Rituma, L., Pretkalnina, L.: Morphological analysis with limited resources: Latvian example. In: Proceedings of the 19th Nordic Conference of Computational Linguistics (NODALIDA) (2013)
7. Pinnis, M., Goba, K.: Maximum entropy model for disambiguation of rich morphological tags. In: Mahlow, C., Piotrowski, M. (eds.) SFCM 2011. CCIS, vol. 100, pp. 14–22. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23138-4_2
8. Virpioja, S., Smit P., Grönroos, S.-A., Kurimo, M.: Morfessor 2.0: Python implementation and extensions for Morfessor baseline. In: Aalto University publication series SCIENCE + TECHNOLOGY, 25/2013, Aalto University (2013)
9. Jurafsky, D., Martin, J.H.: Speech and Language Processing, 2nd edn, pp. 184–187. Prentice Hall, Englewood Cliffs (2009)
10. Clifton, A., Sarkar, A.: Combining morpheme-based machine translation with post-processing morpheme prediction. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 32–42 (2011)
11. Mermer, C., Akin, S.: Unsupervised search for the optimal segmentation for statistical machine translation. In: Proceedings of the ACL 2010 Student Research Workshop, Uppsala, Sweden, pp. 31–36 (2010)
12. Pinnis, M., Krišlauks, R., Miks, T., Deksne, D., Šics, V.: Tilde's machine translation systems for WMT 2017. In: Proceedings of the Second Conference on Machine Translation (WMT 2017). Shared Task Papers, Copenhagen, Denmark, vol. 2, pp. 374–381. Association for Computational Linguistics (2017). http://www.aclweb.org/anthology/W17-4737
13. Grönroos, S.-A., Virpioja, S., Smit, P., Kurimo, M.: Morfessor FlatCat: an HMM-based method for unsupervised and semi-supervised learning of morphology. In: Proceedings of the 25th International Conference on Computational Linguistics, Dublin, Ireland, pp. 1177–1185. Association for Computational Linguistics (2014)
14. Sennrich, R., Firat, O., Cho, K., Birch, A., Haddow, B., Hitschler, J., Junczys-Dowmunt, M., Läubli, S., Barone, A.V.M., Mokry, J., Nadejde, M.: Nematus: a toolkit for neural machine translation. In: Proceedings of the Software Demonstrations of the 15th Conference of the European Chapter of the Association for Computational Linguistics, Valencia, Spain, pp. 65–68 (2017)
15. Gehring, J., Auli, M., Grangier, D., Yarats D., Dauphin, Y.: Convolutional sequence to sequence learning. In: Proceedings of the 34th International Conference on Machine Learning, Sydney, Australia, pp. 1243–1252 (2017)
16. Sennrich, R., Birch, A., Currey, A., Germann, U., Haddow, B., Heafield, K., Barone, A.V. M., Williams P.: The University of Edinburgh's neural MT systems for WMT17. In: Proceedings of the Second Conference on Machine Translation. Shared Task Papers, vol. 2, Copenhagen, Denmark (2017)
17. Barone, A.V.M., Helcl, J., Sennrich, R., Haddow, B., Birch, A.: Deep Architectures for Neural Machine Translation (2017). arXiv Preprints: arXiv:1707.07631 [cs.CL]

Effective Online Learning Implementation for Statistical Machine Translation

Toms Miks, Mārcis Pinnis(✉), Matīss Rikters, and Rihards Krišlauks

Tilde, Vienības gatve 75A, Riga 1004, Latvia
{toms.miks,marcis.pinnis,matiss.rikters,rihards.krislauks}@tilde.lv

Abstract. Online learning has been an active research area in statistical machine translation. However, as we have identified in our research, the implementation of successful online learning capabilities in the Moses SMT system can be challenging. In this work, we show how to use open source and freely available tools and methods in order to successfully implement online learning for SMT systems that allow improving translation quality. In our experiments, we compare the baseline implementation in Moses to an improved implementation utilising a two-step tuning strategy. We show that the baseline implementation achieves unstable performance (from −6 to +6 BLEU points in online learning scenarios and over −6 BLEU points in translation scenarios, i.e., when post-edits were not returned to the SMT system). However, our devised two-step tuning strategy is able to successfully utilise online learning capabilities and is able to improve MT quality in the online learning scenario by up to +12 BLEU points.

Keywords: Phrase-based statistical machine translation
Online learning · Dynamic adaptation

1 Introduction

When working on the translation of documents or larger translation projects, it can easily become annoying if machine translation (MT) systems make the same mistakes on words, phrases, or sentences that were corrected by the translator a few segments earlier. To address this issue for MT systems, researchers have developed online learning (OL) methods that allow improving the translation quality during runtime by learning from corrected translations, which are sent back to the MT system from computer-assisted translation (CAT) tools after translators have approved a post-edited translation.

In this work, we analyse the effectiveness of online learning for two language pairs, for which MT online learning has not been previously applied – English-Estonian and English-Latvian. We build upon the implementation by Bertoldi [3], however, we show that the baseline implementation is sub-optimal and in Sect. 5 we propose a better SMT system model weight tuning strategy that allows us to develop systems of higher translation quality. Compared to

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 302–313, 2018.
https://doi.org/10.1007/978-3-319-97571-9_24

related work, we also evaluate the online learning method on large datasets consisting of 20,000 to 60,000 segments that correspond to approximately 375,000 and 475,000 running tokens respectively.

The paper is further structured as follows: Sect. 2 briefly describes relevant related work on online learning, Sect. 3 describes the data used in our experiments, Sect. 4 describes the baseline systems, Sect. 5 proposes the improved tuning strategy, Sect. 7 analyses translation memory (TM) influence in online learning scenarios, and Sect. 8 concludes the paper.

2 Related Work

Online learning has been studied for both statistical machine translation (SMT; [3,12,18]) and neural machine translation (NMT; [20,21,25]). Although NMT can be considered to be state-of-the-art for broad domain MT (as shown by its dominance in the WMT shared tasks in news translation since 2016 [6,7]), there are translation scenarios where NMT still does not out-perform SMT system translation quality. For instance, when dealing with low-resource language pairs or (more or less) controlled languages (or domains with a limited vocabulary), NMT systems have shown not to perform better than SMT systems [23]. As such scenarios (where limited datasets for narrow domains are available) are frequent for localisation service providers, this work will focus on improving the online learning methods for SMT systems.

Related work on online learning for SMT has been previously focused on the development of methods that introduce dynamic translation and language models along the static models of SMT systems and methods that build translation and language models that are designed to be dynamically adapted using post-edited translations. For instance, Bertoldi [3] introduced cache-based models for online learning in the Moses [15] SMT system. Such models have shown to allow improving translation quality in online learning scenarios if the text that needs to be translated contains repetitions (e.g., repeated words, phrases, or even sentences) [4]. There has been also considerable effort spent on identifying methods that optimise hyper-parameters SMT systems with static and dynamic translation and language models. E.g., Mathur and Cettolo [17] in their work used two optimisation methods – the Downhill Simplex Method and the Modified Hill Climbing –, however, they showed limited quality improvements (by just up to 0.5 BLEU points) in their experiments when comparing systems without online learning and with online learning.

Germann [12] describes suffix-array based translation models for the Moses toolkit. In the online learning scenario, the models are supplemented with phrase translations that are extracted from post-edited sentences. Although being a promising method, the author reported that there was insufficient evidence that the method provides better translation quality compared to a baseline system without online learning. [18] introduced a feature into a suffix-array based translation model that indicates whether phrase candidates are added from the post-edited data. The model weights are continuously tuned during online learning in order to learn that the additional feature is important. The authors report a significant (5 BLEU point) improvement in their experiments. A similar method using the cdec [11] SMT system has been proposed by Denkowski et al. [9] where the authors successfully used a suffix-array based translation model and a cache-based language model for SMT system online adaptation. The authors report an MT error reduction of 13%.

The effectiveness of online learning in post-editing projects has been recently analysed by Bentivogli et al. [2] where the authors compared static SMT systems to SMT systems with online learning capabilities. The authors showed that the post-editing effort when using online learning could be reduced by up to 10%. However, the authors experimented on very small datasets consisting of just up to 300 sentences and the same dataset was post-edited by the translators twice - the first time with the static SMT systems and the second time with the adaptive SMT systems. Although there was a one-month time difference between the post-editing sessions, the translator performance may still be affected by remembering the translations to some extent (or at least remembering how the translation was performed).

3 Data

We evaluate the online learning method on datasets from two domains (information technologies (IT) and medicine) and for two language pairs (English-Estonian and English-Latvian). For training of the medical domain systems, we use publicly available data - the parallel corpus of the European Medicines Agency (EMEA; [24]) that primarily consists of drug (medicine) descriptions. For training of the IT domain systems, we use a collection of publicly available corpora (e.g., the Microsoft User Interface Translations parallel corpus [19]) and proprietary corpora (e.g., software documentation, user interface strings, etc.). Note that the IT domain corpora are of broader coverage in terms of vocabulary and writing styles than the medical domain corpus, which is mostly constructed from medicine descriptions. The training data statistics are summarised in Table 1. It is evident that both domains present different MT scenarios - the medical domain scenario is a low resource and narrow domain scenario, whereas the IT domain scenario is a high resource and broad domain scenario.

Table 1. Training data statistics

	IT domain		Medical domain
	English-Estonian	English-Latvian	English-Latvian
Unique parallel sentence pairs	9,059,100	4,029,063	325,332
Unique in-domain monolingual sentences	34,392,322	1,950,266	332,652
Unique broad domain monolingual sentences	-	2,369,308	-
Tuning data	1,990	1,837	2,000

For evaluation of the online learning method, we use data from two large post-editing projects - a commercial post-editing project for a private customer in the IT domain (for English-Latvian and English-Estonian), and a research post-editing project in the medical domain (for English-Latvian). The post-edited data for the experiments in the medical domain have been produced within the QT21 project[1] [22]. The IT domain data were prepared by post-editing MT translations of software documentation, user interface strings, and (IT product related) marketing texts within the MemSource[2] web-based computer-assisted translation (CAT) tool. The translations were provided by a phrase-based SMT system that was trained on a similar corpus as the training data that were used in our experiments. The medical domain data were prepared by post-editing MT translations of medicine descriptions from the EMEA home page using the Post-editing Tool (PET; [1]). The SMT system that prepared the initial translations was trained on the same training data that were used in our experiment. The training data do not include documents from the online learning evaluation set, however, there may be individual sentences that appear in the training data (we believe, that this allows to better simulate real-life translation situations where some sentences tend to be repetitive). The post-edited data statistics are given in Table 2.

Table 2. Post-edited data statistics

	IT domain		Medical domain
	English-Estonian	English-Latvian	English-Latvian
Segments	60 630	27 122	20 286
Tokens	475 295	166 350	374 914

4 Baseline Implementation

We started our experiments by training baseline SMT systems and SMT systems with baseline dynamic learning models. All MT systems were trained using the Moses SMT system on the Tilde MT[3] platform [26]. For word alignment, we used fast-align [10]. All SMT systems feature 7-gram translation models and the *wbe-msd-bidirectional-fe-allff* reordering models. The systems have either one or two language models (depending on the availability of broader domain data) that were trained using KenLM [14]. For English-Latvian, we trained 5-gram language models and for English-Estonian (due to a significantly larger monolingual corpus) - 4-gram language models. The systems were tuned using MIRA [13] on the respective tuning datasets.

[1] More information about the QT21 project can be found online at http://www.qt21.eu/.

[2] www.memsource.com.

[3] www.tilde.com/mt.

The online learning set-up is based on the implementation by Bertoldi [3]. First, the static SMT system's models are trained, after which a dynamic translation model and a dynamic language model are added to the SMT system. The system's model (both static and dynamic) log-linear weights are tuned using MIRA by iteratively translating the tuning dataset using the online learning procedure. The online learning procedure during translation is as follows:

1. The SMT system receives a translation request to translate a sentence.
2. The sentence is translated by the SMT system and the translation is sent to a CAT tool.
3. The translation is post-edited by a translator in the CAT tool.
4. The post-edited translation together with the source sentence is sent back to the SMT system to perform online learning.
5. The SMT system performs word alignment between the source sentence and the post-edited sentence using fast-align. For this, we use the fast-align model acquired during training of the SMT system.
6. The SMT system extracts parallel phrases consisting of 1-7 tokens using the Moses phrase extraction method [16] that is implemented in the Moses toolkit.
7. The extracted phrases are added to the dynamic translation and language models so that, when translating the next sentence, the system would benefit from the newly learned phrases. Phrases that are added to the dynamic models are weighted according to their age (newer phrases have a higher weight) using the hyperbola-based penalty function [3]. A maximum of 10,000 phrases is kept in the dynamic models.

In our experiments we distinguish three types of translation scenarios:

1. The baseline scenario uses a standard SMT system with no dynamic models.
2. The *OL−* scenario uses an SMT system with dynamic models, however, post-edited translations are not sent back to the SMT system for online learning. This means that the dynamic models will always stay empty. The goal of this scenario is to validate whether SMT systems with dynamic models are able to reach baseline translation quality in situations when some CAT tools are not able to or do not provide functionality that allows returning post-edited translations back to the SMT system.
3. The *OL+* scenario uses an SMT system with dynamic models and after translation of each sentence, the post-edited translation is sent back to the SMT system for online learning.

Note that having a translator ready for every experiment is expensive and time-consuming. Therefore, online learning is evaluated in a simulated online learning scenario where instead of the (dynamic) post-edits, which should be prepared by a translator when using an online learning enabled SMT system, we use (static) post-edits that were collected in the post-editing projects, where translators used a static SMT system (without online learning capabilities).

The baseline systems were evaluated using BLEU on the full post-edited datasets. Evaluation results are given in Table 3. For English-Estonian we trained only the baseline SMT system, because we started our experiments for English-Latvian and validated only the best performing set-up for English-Estonian. The *OL−* system results show that the addition of the dynamic models negatively impacted translation quality even though the dynamic models were kept empty (for more details on why this happened, see Sect. 5). However, the translation quality does improve for the broader IT domain *OL+* system by 6 BLEU points when compared to the baseline. This means that the negative effects introduced by the dynamic models can be overcome by online learning over time. For the medical domain, the quality of the *OL+* system dropped by over 7 BLEU points. We believe that this may be caused by the high quality of the baseline system and the fact that the narrow domain data are well represented in the training dataset.

Table 3. Baseline system evaluation results

System	IT domain		Medical domain
	English-Estonian	English-Latvian	English-Latvian
Baseline	26.80 ± 0.17	26.42 ± 0.23	76.78 ± 0.17
OL−	-	19.91 ± 0.20	70.27 ± 0.20
OL+	-	32.42 ± 0.30	69.53 ± 0.22

5 Two-Step Tuning

The *OL−* systems showed a significant drop in translation quality when the post-edited sentences were not used for online adaptation (i.e., if the dynamic models were kept empty). Therefore, we looked into the tuning process and identified that when tuning all log-linear model weights together (i.e., the static model and the dynamic model weights), the tuning method did not find optimal weights for the static models. This led to the significant drop in translation quality of over 6 BLEU points for both the medical and IT domain datasets.

To address the issue, we devised a two-step tuning procedure where the static and dynamic model weights were tuned separately. The tuning method works as follows:

1. First, the static model weights are tuned using MIRA in a standard translation scenario (without online learning).

2 Then, the dynamic model weights are tuned using MERT [5] in an online learning scenario using the pre-trained static model weights. The static model weights are kept unchanged. During dataset analysis, we observed that the repetition rates [8] for the tuning and test datasets differ. We artificially increase the repetition rate in the tuning dataset to more closely match that of the test dataset, which, as our experiments showed (see Sect. 6), increases system performance. As the tuning datasets are random held-out datasets from the training data, we duplicated every n^{th} sentence in order to introduce repetitiveness in the data. For the IT domain experiments, every fourth

sentence was duplicated, and for the medical domain experiments, every sixteenth sentence was duplicated. The duplication rate differs as the medical domain data are much more narrow and they contain higher repetitiveness before duplication than the IT domain data.

The two-step tuning procedure ensures that even if the dynamic models and online learning will not be used (for instance, if a particular CAT tool that a translator uses to post-edit MT translations is not able to send the post-edited translations back to the SMT system), the SMT system will perform as good as the baseline system without any dynamic models.

The evaluation results in Table 4 show that the system quality in both scenarios (*OL*− and *OL*+) is improved by a large margin over the respective baseline systems (see Table 3). For instance, the quality of the English-Latvian IT domain *OL*+ system gained 6.17 BLEU points over the respective baseline. Although in the medical domain the *OL*+ system did not show an improvement in comparison to the *OL*− system, the quality decrease (−0.55 BLEU points) is much lower compared to the baseline *OL+* system's quality decrease (−7.25 BLEU points).

Table 4. Evaluation results for systems with two-step tuning

System	IT		Medical
	English-Estonian	English-Latvian	English-Latvian
Baseline	26.80 ± 0.17	26.42 ± 0.23	76.78 ± 0.17
OL−	26.80 ± 0.17	26.42 ± 0.23	76.78 ± 0.17
OL+	31.45 ± 0.20	38.59 ± 0.31	76.23 ± 0.19

6 Text Repetitiveness in the Tuning Dataset

The potential benefit of online learning depends on how much repetition is evident in the translatable content. Text repetition is also necessary for tuning data in order to successfully tune the dynamic translation and language model weights of the SMT systems. Therefore, in this section, we analyse the level of text repetitiveness required in the tuning dataset to achieve higher MT quality in online learning scenarios. We limit these experiments to the English-Latvian language pair.

As explained above, to simulate text repetitiveness in the tuning dataset, we duplicate every n^{th} (first, fourth, eight, or sixteenth) sentence pair, thereby creating four different tuning datasets. Using these datasets to tune the dynamic model weights, different weight values were identified (giving more or less strength to the dynamic models). Then, evaluation datasets were translated in the *OL+* scenario.

Evaluation results in Table 5 indicate that for the medical domain, better results are attained when every 16^{th} sentence in the tuning dataset was repeated. On the other hand, for the IT domain, the best results were attained when repeating every fourth sentence.

Table 5 provides also scores that analyse how much repetition is present in the tuning datasets and for reference also for the evaluation datasets using two text repetitiveness metrics. The first is the Repetition Rate (RR) metric [8]. The RR metric calculates text repetitiveness by analysing the number of n-grams (from 1 to 4 tokens) repeated in the text. Our observations showed that the repetition of 4-grams in the evaluation data was relatively low. Therefore, we devised a modified text repetitiveness metric – RR1 – that considers only unigrams, bigrams and trigrams. RR calculates text repetitiveness by analysing the text as a whole, thereby ignoring sentence boundaries. Since MT systems operate on a sentence-level, we restricted the RR1 metric to count n-gram repetitiveness only within sentence boundaries (and not between sentences).

The results show that for medical domain data, the highest MT quality is achieved when the repetitiveness in the tuning data is similar to the repetitiveness in the evaluation data (according to both metrics – RR and RR1). The situation slightly differs for IT domain data. It is evident that the tuning data and evaluation data RR scores differ for the configuration that achieves the best results. However, the RR1 scores of the tuning data for the best performing configuration are the most similar to the RR1 scores of the evaluation data.

The results indicate that in order to achieve the highest MT quality in online learning scenarios, the text repetitiveness according to the RR1 metric in tuning data has to be similar to the text repetitiveness in the evaluation data. The tendency is clearer when plotting the results in a chart (see Fig. 1).

Table 5. Translation quality results for English-Latvian in the *OL+* scenario for different levels of text repetitiveness in the tuning datasets.

Experiment	Medical domain			IT domain		
	RR1	RR	BLEU	RR1	RR	BLEU
Evaluation data	0.13	0.11	-	0.31	0.20	-
Tuning data – 100% repetitiveness	0.51	0.85	75.98 (75.61-76.33)	0.51	0.85	36.01 (35.42-36.63)
Tuning data – 25% repetitiveness	0.27	0.31	76.17 (75.81-76.54)	0.26	0.30	38.59 (38.01-39.21)
Tuning data – 12.5% repetitiveness	0.19	0.21	76.18 (75.83-76.54)	0.19	0.20	38.38 (37.77-38.95)
Tuning data – 6.25% repetitiveness	0.14	0.16	76.23 (75.88-76.6)	0.14	0.15	38.38 (37.79-39.02)

7 Translation Memory Influence

The Tilde MT platform provides a translation memory feature for all MT systems that send post-edited translations back to the platform. This means that during online learning scenarios, for sentences, for which full match sentences can be found, the translations are looked-up in the translation memory of the

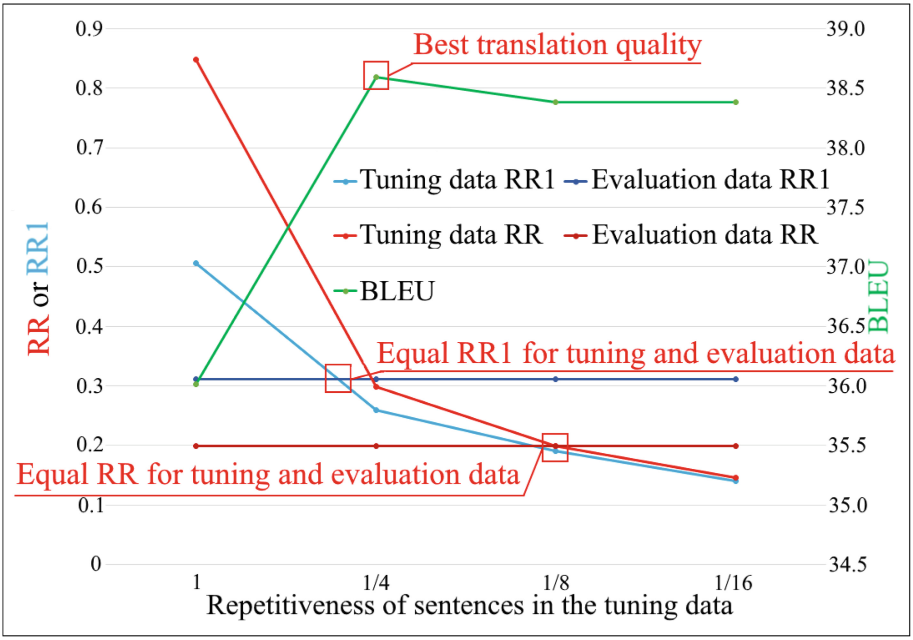


Fig. 1. Difference of translation quality for different text repetitiveness levels in the tuning data

SMT system. This means that it is important to identify, how large improvement in online learning scenarios is obtained from the translation memory alone and how large improvement can be attributed to the online learning method itself.

When analysing the post-edited data, we identified that there is a sentence level repetitiveness of 17.8% and 15% in the English-Latvian and English-Estonian IT domain post-edited datasets respectively. The repetitive segments on average consist of 2.1 and 2.5 English words in the respective datasets. All unique segments on average consist of 6.3 and 7.1 words respectively, which means that the repetitive segments are mostly short phrases (e.g., repetitive menu item, button, and label titles, etc.). The repetitiveness in the medical domain corpus was 0% due to how the data for post-editing was prepared, therefore, this analysis was not performed on the medical domain dataset.

To analyse the impact of the translation memory on the translation quality, we performed an additional experiment where post-edited sentences were used to fill the translation memory, but the online learning functionality was disabled. The results of the experiment are given in Table 6. It is evident that the translation memory accounts for 2 to 2.5 BLEU points for both language directions. However, the improvement from online learning is still substantial (9.62 and 2.58 BLEU points for English-Latvian and English-Estonian respectively over using just the translation memory). The cumulative improvement of 12.17 and 4.65 BLEU points for English-Latvian and English-Estonian respectively shows that

Table 6. Individual and cumulative impact of the translation memory and online learning on the translation quality using the IT domain datasets

System	English-Estonian (BLEU)	English-Latvian (BLEU)
Baseline	26.80±0.17	26.42±0.23
Baseline + translation memory (improvement)	28.87±0.20 (+2.07)	28.97±0.26 (+2.55)
OL+ + translation memory (improvement)	31.45±0.20 (+2.58)	38.59±0.31 (+9.62)
Total improvement	**+4.65**	**+12.17**

in order to achieve the best results, it is beneficial to use both the translation memory and the online learning functionality.

8 Conclusion

In this paper, we described an online learning method for SMT systems based on the implementation by Bertoldi [3] and open source and freely available tools. We showed that the baseline implementation did not allow to improve SMT system quality due to sub-optimal tuning performance when adding the dynamic models to the SMT system. To address this issue, we devised a two-step tuning method, which, first, identifies good weights for the SMT system's static models and only then tunes the dynamic model weights in an online learning set-up.

Our experiments showed that the improved online learning method in combination with a translation memory allowed to increase IT domain SMT system quality from +4.65 (for English-Estonian) up to +12.17 (for English-Latvian) BLEU points. We also showed that although for narrow domain systems of very high quality (i.e., for systems of over 75 BLEU points) the online learning method did not show an improvement, the drop in quality is fairly minimal (just 0.55 BLEU points).

Finally, we analysed also the impact of text repetitiveness in the tuning dataset on the MT quality in online learning scenarios. The results showed that in order to achieve the highest MT quality, it is important for the tuning dataset to feature a level of text repetitiveness that matches the natural text repetitiveness of the data to be translated.

We believe that the findings of the paper will help other researchers and SMT system developers to successfully develop online learning systems that allow improving SMT system quality.

Acknowledgements. We would like to thank Tilde's Localization Department for the hard work they did to prepare the post-edited data analyses in this work. The research has been supported by the ICT Competence Centre (www.itkc.lv) within the project "2.2. Prototype of a Software and Hardware Platform for Integration of Machine Translation in Corporate Infrastructure" of EU Structural funds, ID n° 1.2.1.1/16/A/007.

References

1. Aziz, W., De Sousa, S.C., Specia, L.: Pet: a tool for post-editing and assessing machine translation. In: Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC 2012), pp. 3982–3987 (2012)
2. Bentivogli, L., Bertoldi, N., Cettolo, M., Federico, M., Negri, M., Turchi, M.: On the evaluation of adaptive machine translation for human post-editing. IEEE/ACM Trans. Audio Speech Lang. Process. (TASLP) **24**(2), 388–399 (2016)
3. Bertoldi, N.: Dynamic models in Moses for online adaptation. Prague Bull. Math. Linguist. **101**, 7–28 (2014). https://doi.org/10.2478/pralin-2014-0001.Brought
4. Bertoldi, N., Cettolo, M., Federico, M.: Cache-based online adaptation for machine translation enhanced computer assisted translation. In: Proceedings of the XIV Machine Translation Summit, pp. 35–42 (2013)
5. Bertoldi, N., Haddow, B., Fouet, J.B.: Improved minimum error rate training in Moses. Prague Bull. Math. Linguist. **91**(1), 7–16 (2009)
6. Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huang, S., Huck, M., Koehn, P., Liu, Q., Logacheva, V., et al.: Findings of the 2017 conference on machine translation (wmt17). In: Proceedings of the Second Conference on Machine Translation, pp. 169–214 (2017)
7. Bojar, O., Chatterjee, R., Federmann, C., Graham, Y., Haddow, B., Huck, M., Yepes, A.J., Koehn, P., Logacheva, V., Monz, C., et al.: Findings of the 2016 conference on machine translation. In: ACL 2016 First Conference on Machine Translation (WMT 2016), pp. 131–198. The Association for Computational Linguistics (2016)
8. Cettolo, M., Bertoldi, N., Federico, M.: The repetition rate of text as a predictor of the effectiveness of machine translation adaptation. In: Proceedings of the 11th Biennial Conference of the Association for Machine Translation in the Americas (AMTA 2014), pp. 166–179 (2014)
9. Denkowski, M., Lavie, A., Lacruz, I., Dyer, C.: Real time adaptive machine translation for post-editing with cdec and transcenter. In: Proceedings of the EACL 2014 Workshop on Humans and Computer-Assisted Translation, pp. 72–77 (2014)
10. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL HLT 2013), Atlanta, USA, pp. 644–648, June 2013
11. Dyer, C., Weese, J., Setiawan, H., Lopez, A., Ture, F., Eidelman, V., Ganitkevitch, J., Blunsom, P., Resnik, P.: cdec: a decoder, alignment, and learning framework for finite-state and context-free translation models. In: Proceedings of the ACL 2010 System Demonstrations, pp. 7–12. Association for Computational Linguistics (2010)
12. Germann, U.: Dynamic phrase tables for machine translation in an interactive post-editing scenario. In: Proceedings of AMTA 2014 Workshop on Interactive and Adaptive Machine Translation, pp. 20–31 (2014)
13. Hasler, E., Haddow, B., Koehn, P.: Margin infused relaxed algorithm for moses. Prague Bull. Math. Linguist. **96**, 69–78 (2011)
14. Heafield, K.: KenLM: faster and smaller language model queries. In: Proceedings of the Sixth Workshop on Statistical Machine Translation, No. 2009, pp. 187–197. Association for Computational Linguistics (2011)

15. Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., Herbst, E.: Moses: open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions, ACL 2007, Stroudsburg, PA, USA, pp. 177–180. Association for Computational Linguistics (2007). http://dl.acm.org/citation.cfm?id=1557769.1557821
16. Koehn, P., Och, F.J., Marcu, D.: Statistical phrase-based translation. In: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology, vol. 1, pp. 48–54. Association for Computational Linguistics (2003)
17. Mathur, P., Cettolo, M.: Optimized MT online learning in computer assisted translation. In: IAMT 2014-AMTA 2014 Workshop on Interactive and Adaptive Machine Translation, pp. 32–41 (2014)
18. Mathur, P., Cettolo, M., Federico, M., Kessler, F.F.B.: Online learning approaches in computer assisted translation. In: WMT@ACL, pp. 301–308 (2013)
19. Microsoft: Translation and UI strings glossaries (2015)
20. Peris, Á., Casacuberta, F.: Online learning for effort reduction in interactive neural machine translation (2018). arXiv preprint: arXiv:1802.03594
21. Peris, A., Cebrián, L., Casacuberta, F.: Online learning for neural machine translation post-editing (2017). arXiv preprint: arXiv:1706.03196
22. Pinnis, M., Kalniņš, R., Skadiņš, R., Skadiņa, I.: What can we really learn from post-editing? In: Proceedings of the 12th Conference of the Association for Machine Translation in the Americas (AMTA 2016). MT Users, vol. 2, Austin, USA, pp. 86–91. Association for Machine Translation in the Americas (2016)
23. Skadiņa, I., Pinnis, M.: NMT or SMT: case study of a narrow-domain English-Latvian post-editing project. In: Proceedings of the Eighth International Joint Conference on Natural Language Processing. Long Papers, vol. 1, pp. 373–383 (2017)
24. Tiedemann, J.: News from OPUS - a collection of multilingual parallel corpora with tools and interfaces. Recent Adv. Nat. Lang. Process. **5**, 237–248 (2009)
25. Turchi, M., Negri, M., Farajian, M.A., Federico, M.: Continuous learning from human post-edits for neural machine translation. Prague Bull. Math. Linguist. **108**(1), 233–244 (2017)
26. Vasiļjevs, A., Skadiņš, R., Tiedemann, J.: LetsMT!: a cloud-based platform for do-it-yourself machine translation. In: Proceedings of the ACL 2012 System Demonstrations, Jeju Island, Korea, pp. 43–48. Association for Computational Linguistics, July 2012

Investigation of Text Attribution Methods Based on Frequency Author Profile

Polina Diurdeva[1](✉) and Elena Mikhailova[1,2](✉)

[1] Saint Petersburg State University, Saint Petersburg, Russia
polina.duedeva@yandex.ru
[2] ITMO University, Saint Petersburg, Russia
e.mikhaylova@spbu.ru

Abstract. The task of text analysis with the objective to determine text's author is a challenge the solutions of which have engaged researchers since the last century. With the development of social networks and platforms for publishing of web-posts or articles on the Internet, the task of identifying authorship becomes even more acute. Specialists in the areas of journalism and law are particularly interested in finding a more accurate approach in order to resolve disputes related to the texts of dubious authorship. In this article authors carry out an applicability comparison of eight modern Machine Learning algorithms like Support Vector Machine, Naive Bayes, Logistic Regression, K-nearest Neighbors, Decision Tree, Random Forest, Multilayer Perceptron, Gradient Boosting Classifier for classification of Russian web-post collection. The best results were achieved with Logistic Regression, Multilayer Perceptron and Support Vector Machine with linear kernel using combination of Part-of-Speech and Word N-grams as features.

Keywords: Author attribution · Text classification
Frequency author profile

1 Introduction

Author Identification (AI) task has become more interesting due to publishing a large amount of text data on the Internet. Daily in social networks, blogs, e-journals enormous number of anonymous authors publish a lot of materials. Some of the materials sometimes need to be deanonymized. The task of identifying authors of web texts is more complex than the one for literary texts to a number of reasons. First, such materials usually are not long enough. Second, unlike literary texts, where authors have a more stable writing style and try to stick to it in order to stand out and be recognized, web materials are usually published spontaneously and the authors are less concerned about having an individual style.

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 314–327, 2018.
https://doi.org/10.1007/978-3-319-97571-9_25

In current work we focused on classification approach for solving AI task for Russian texts. Classification of texts implies assignment of each anonymous text to a certain class. Texts of the same class belong to the same particular author. Those classes are generated from the training text dataset where authorship of each text is known in advance [16].

The choice of a method used when assigning texts to classes is an issue. Typically Machine Learning methods and natural language processing are heavily used as they allow to effectively classify and process data. For example, some of the most popular methods for the problem under consideration are Naive Bayes and Support Vector Machine. These methods actually show quite high accuracy for different languages. However, picking the right methods is not a panacea.

In addition, regardless of the picked algorithm, the text classification task requires solving several problems of text analysis simultaneously: feature extraction, feature selection, feature representation, and preprocessing of texts. The main stages are briefly described below.

Feature Extraction: The primary problem is to search for representative features in the text. Many works are devoted to the problem of selection of such signs. The purpose of the problem of finding informative properties of a text, (writing style) is to improve the accuracy of the final algorithm. Properties or characteristics of the text can be divided into 5 main groups: lexical, grammatical, syntactic, structural and substantive. In this study, we examined such features as N-grams of Parts-of-Speech, N-grams of Words and their combinations.

Feature Representation: After fixing the features of the text that will in some way represent the analyzed texts, it is necessary to convert the combination of these characteristics in the feature vector. Thus, the second problem of text analysis is constructing a vector representation of text. The solution to this problem can be a binary representation, a frequency representation, or, for example, a $TF-IDF$ representation. When testing algorithms, we considered the frequency model and the $TF - IDF$ model, however, in the course of experiments it was found that the results for the frequency model are lower on 1–3%.

Feature Selection: Often, the space of extracted features has a high dimension. This, first of all, leads to an increase in the complexity of the classification. There are several techniques to reduce the dimensionality and select more useful features: top-k, chi-square, information gain, using Pearson correlation and etc.

It should be noted that performance of each method may also depend on the language of texts it is applied to. In the current paper we investigate on the applicability of 8 popular classification methods for solving *AI* task on *Russian* texts taken from web. A remarkable feature of the Russian language is that sentence structure is very flexible. Moreover, one Russian word might have a lot of grammatical forms. These and other properties might affect the performance of the algorithms therefore we decided to examine how they will work on short Russian web texts in more detail.

This paper is organized as follows: Sect. 2 describes the related work; Sect. 3 presents experimental text corpus and steps of text processing; Sect. 4 presents the experimentation and results; in Sect. 5 the paper concludes.

2 Related Works

Majority of state-of-art works on the text analysis are based on Machine Learning methods. In work [2] authors solve problem of Author Identification comparing Random Forest, Logistic Regression and SVM. Unigrams, bigrams and latent semantic features on document word space were measured. Research was submitted for PAN-CLEF 2014 Author Attribution task for English, Dutch, Greek and Spanish languages. More precise result was obtained for Random Forest with average accuracy 80%. Research [7] is one of the first works where the performance of Random Forest in problem of Author Attribution was evaluated. Authors of this study interpret a text as a union of different features like word frequencies and numbers, frequency of word lengths, frequency of N-grams, hapax and Yules richness and others. They achieve reasonable result on PAN12 dataset comparable with one obtained on PAN-CLEF 2012. Comparison of SVM (Sequential Minimal Optimization) and Random Forest performance was made in [4].

SVM and its modifications are well-proven text classification methods applied not only to the problem of Authorship Identification. In a research [6] authors dealt with big corpus of Lithuanian Internet comments containing extremely short texts: 20 26 tokens per text in average. They explored the impact of most popular feature types: bag of words, word lemmas, word level tetra-grams and tested SVM (SMO), Naive Bayes and Similarity-based approach. SVM outperformed all other explored algorithms. In work [9] authors also conducted comparison of SVM and Naive Bayes algorithms for AI task for Arabic texts. This work examined short texts of 10 authors. As features word N-grams and character N-grams were selected. The interesting results was that on the opposite to [6] best result equal to 96% was obtained on Naive Bayes algorithm with word unigrams.

Logistic Regression method is less investigated in the sphere of text analysis compared with SVM or Naive Bayes. Authors of work [13] applied it for multiclass text classification with Part-of-Speech tagging approach. Amazon customer product reviews were examined. The highest average accuracy was equal to 59% in case of extracting and combining 1,2,3-pos-grams.

Applicability of Machine Learning algorithm for Russian language is a less investigated direction. In [17] authors tested a stack of classification algorithms on imbalanced corpora of Russian web-posts. Texts were presented as combination of various lexical, semantic-structural and metadata features. Comparison of six algorithms: SVM (SMO), Multilayer Perceptron, Decision Tree, Random Forest, Logistic Regression and Naive Bayes was performed. Random Forest algorithm with the accuracy approximately equal to 60–75% particular experiment setting demonstrated the best performance.

It is no doubt that the accuracy of the classification is highly dependent on the extracted characteristics from texts. It was showed in work [12] that Part-of-Speech N-grams of variable length can be successfully used to identify authors writing style. The proposed computational method based on Pearson correlation and using only 50 most frequent sequential N-grams achieves more than 70% accuracy. The limitation of this study is that experimental dataset has long enough texts: considered books had 3300 sentences in average that was much more than common web-posts have.

In [15] significant research was conducted for analysis of the performance of over 100 variants of 5 filter feature selection methods. This study investigated Naive Bayes, Rocchio, K-Nearest Neighbor and SVM on two corpora (Reuters 21578 and part of RCV1). It was demonstrated that exclusion of rare words and using chi-square or information gain filters with document frequency help to increase the accuracy of classification.

Besides Author Identification problem Machine Learning algorithms also enable to achieve accurate results for Author Profiling [10,14], Plagiarism Detection [5].

Extensive variety of research that has been done so far shows that performance of applied methods are very dependent on language, dataset properties, selected features.

3 Text Collection and Representation of Text

3.1 Text Collection

Plenty of studies regarding to text analysis take into account experiments on English corpora. In current research experiments were performed on collection of Russian texts. The corpus was formed of blog-posts selected from site of radio station Echo of Moscow. Collection contains texts of 17 authors. Each author had a set of 40 texts. In the corpus the texts on politic topic numerically predominated over the texts on economic and culture. The length of the texts had a large scatter. The shortest text consisted of 32 words and the longest of 4261 words. Most of the collection consisted of texts whose length ranges from 100 to 1000 words. In Table 1 the average, maximum and minimum number of words in texts for each author are presented.

3.2 Text Representation

In this section we discuss steps of the AI problem which was touched upon a bit earlier. In current study experiments were conducted on several combination of feature extraction and feature selection approaches. The goal of such approaches is to improve the accuracy of the final algorithm by choosing more informative features and a more appropriate way to present them, to reduce computational complexity of the solution or both.

3.3 Pre-processing

This step usually implies markup tags and punctuation mark removal, stop-words exclusion, text normalization, lemmatisation, etc.

Table 1. Statistical information of text collection

Author id	Avg length	Min length	Max length
a1	518	140	1526
a2	294	72	910
a3	792	132	2306
a4	404	32	1524
a5	600	45	1328
a6	399	121	1373
a7	348	80	1055
a8	470	341	567
a9	550	90	1456
a10	658	146	2603
a11	852	39	2502
a12	509	191	726
a13	190	40	3093
a14	531	100	1157
a15	676	155	1677
a16	675	275	1734
a17	560	39	4261

Our pre-processing includes cleaning web-posts of markup tags. In case of word N-gram model all words are brought into normal form of the singular nominative case. Moreover, we remove all marks of punctuation. Conversely for POS N-gram model we save all punctuation marks. It was done with assumption that frequent occurrence of such marks or their occurrence after particular Part-of-Speech also may be an outstanding characteristic of author's style. In addition, in [3] experiments showed that preserving of punctuation does not impair the quality of the classification and even in some cases the accuracy can be significantly improved. Also we did not exclude stop words for the same reasons.

However, there are no doubts that some words can indeed make no sense for classification due to equally distributed in the texts of all authors. We delegate solving this problem to techniques used in the following steps of the text presentation.

Feature Extraction. Whatever the purpose is, the text analysis process is a complex challenge. The most important subtask is to pick out representative features or characteristics from documents and transform raw data into numerical features suitable for applying the classification algorithm. In current work three N-gram models were for comparison.

Part-of-Speech (POS) N-gram Model. POS tags were used as features extracted from texts. Pymorphy2 [8] was used as a tool for morphological analysis.

This library allows determining grammatical properties of Russian words including POS tags. In total, Pymorphy2 can refer a word to one of 17 Parts-of-Speech such as noun, adverb, verb, etc. It is assumed that the author's writing style can be expressed in a tendency of usage of certain Parts-of-Speech or their sequences. We did not limit the set of POS tags provided by Pymorphy2. After that each word or punctuation character is associated with a corresponding marker, the text is a sequence of these tags.

The next task is to transform a sequence of tags into a vector of features using a specific computational algorithm. Primarily, sequential N-grams of POS tags were extracted from texts where N was variable length from 1 to 3. To count weighted frequency of N-grams the Term Frequency Inverse Document Frequency ($TF-IDF$) model was applied. The advantage of this model is that it takes into account the significance of the feature for a particular document and for the document space as a whole. The purpose of using $TF-IDF$ is to diminish influence of features which appears frequently in the whole collection of documents because it may mean that such features inherent the specificity of language but not a author's writing style. The equation of $TF-IDF$ Frequency is presented in (1).

$$TF-IDF(d,t) = (1 + \log tf(t,d)) * idf(t) \tag{1}$$

where tf is number of time term occurs in document.

$$idf(t) = \log \frac{n_d}{df(d,t)} + 1 \tag{2}$$

where n_d is the total number of documents, and $df(d,t)$ is the number of documents that contain term t.

Word N-gram Model. The second model used N-gram model over word space of documents. N was varying from 1 to 2 so separate word and sequences of two words were considered as features. In addition, all the words were brought into the normal form using Pymorphy2, for example, nouns, adjectives, adverbs etc. were brought into singular (if any) nominative case, while verbs were brought into infinitive. Pronouns, conjunctions, particulars and so on were not transformed. For this model $TF-IDF$ method was also used to compute occurrence frequency of word N-grams.

Combine N-gram Model. The next model was build as combination of POS and Word N-gram models. We tried to find a balance between number of POS N-grams and Word N-grams that would be included to feature vector.

Feature Selection. Feature selection techniques allow to scale down the size of feature vectors and remove the noise which constitute irrelevant features. Two techniques were used in current paper.

The first approach reduces dimensionality by sorting features according to $TF-IDF$ values and taking top-k. The experiments with different values of k were performed to find the optimal value.

The second one is chi-squared method which computes chi-squared stats between each feature and class. The top-k features with the highest score are selected. The method eliminates those feature that are most likely to be independent of a class.

4 Classifiers

We explored 8 different classifiers:

1. Support Vector Machine
2. Multinomial Naive Bayes
3. Logistic Regression
4. K-nearest Neighbors
5. Decision Tree
6. Random Forest
7. Multilayer Perceptron
8. Gradient Boosting Classifier

For this purposes scikit-learn [11] Python library was used. We perform grid searching with sets of basic parameters for each algorithm to find more appropriate ones. Below we elaborate on each algorithm briefly.

Support Vector Machine (SVM). SVM is often used in the task of categorizing texts due to its properties. SVM copes with handling high dimension sparse space which is the space of feature vectors. We tested implementation of Linear Support Vector Classification which assumes linear kernel. The Linear kernel is computationally cheaper than other kernels and previous studies reported good results for text classification task.

Naive Bayes Multinomial. Multinomial NB model is probabilistic learning method that effectively applied for document classification. It is also highly applicable to data of high dimension. It is often used to solve text analysis problems because of its simplicity, reliability and speed. It works particularly well on small datasets or short documents.

Logistic Regression. Logistic regression is a linear classifier. Logistic regression works well with a large amount of data and performs well used for classifying long documents. One of its strengths is that Logistic Regression gracefully deals with correlated features.

K-nearest Neighbors. The algorithm is known to be successfully applicable in many domains, however it also has some considerable disadvantages. Well-known disadvantage of the algorithm is that it is poorly resistant to noise. Finding k nearest Neighbors might also become costly when a lot of features are used. In out experiments we set k to 10.

Decision Tree. Decision Tree has a few advantages for AI task but we anyway included in our bunch of experiments just for comparison with others. Their ability to select features with the most discriminatory power is a very useful property of the Decision Tree. Decision Tree is very fast in testing phase after building the tree, but training phase can take a reasonable amount of time. Moreover, this algorithm has a drawback of being sensitive to irrelevant features. Despite the fact that Decision Tree can handle categorical and continuous features, usage of large amount of continuous features increases the cost of computing and significantly decrease the quality of classification [1].

Random Forests. Random Forests are an ensemble learning method that constructs a number of decision trees at training time. Random forest is a very reliable, robust and versatile method, however its no usually appropriate for high dimensional sparse data. Anyway it demonstrates considerable results in AI tasks in other works. During our experiments we set *n_estimator* to 500.

Multilayer Perceptron. The logistic activation function was used. We used 1 hidden layer with 100 neurons. Multilayer Perceptron has a remarkable property of being able to find hidden relations by itself in complex noisy data, that are extremely hard to be noticed by other machine learning techniques. In addition, Multilayer Perceptron is computationally heavy one and requires significantly high processing power.

Gradient Boosting Classifier. Today, boosting is one of the most powerful recognition algorithm. Boosting over decisive trees might be one of the most effective option. However, this algorithm is quite time-consuming, slow and difficult to configure. We set *n_estimator* (stages to perform) equal to 500 and *max_depth* (the maximum depth of the individual regression estimators) equal to 5.

4.1 N-Fold Cross Validation

N-cross validation technique is commonly used to evaluate predictive model when there is not enough data in a corpus to divide it into training and test data set and to obtain a statistically significant accuracy of a model. We set N equal to 10. It means that collection of documents was randomly divided into 10 equally sized disjoint parts where 9 parts were used for training and the remaining one for model testing. Moreover, stratified selection was applied to obtain approximately the same percentage of samples of each target class as the complete set in each fold. The evaluating model procedure are repeated 10 times and each of 10 parts were used as a test data exactly once. It allows us to examine how widely the performance varies across different training sets. If we get very similar scores for all N training sets, then we can be fairly confident that the score is accurate. As a result it enables to get 10 values of accuracy and compute average accuracy for dataset.

4.2 Experiments

We conducted experiments with several parameter configurations for models that have been considered previously.

Experiments were conducted using k-fold cross validation with $k = 10$, therefore we evaluated the performance of the classification algorithms by calculation of average accuracy and average F1-score (macro) over results obtained on 10 stratified samples. The formula according to which the accuracy was computed is presented in Eq. 3. In Eq. 4 the formula for F1 (macro) is presented. F1-score combines estimation of precision and recall and can be interpreted as their weighted average. To obtain the total quality estimates of the classification by classes macro averaging is introduced.

$$accuracy = \frac{Number\ of\ correctly\ identified\ documents}{Total\ number\ of\ documents} \tag{3}$$

$$f1 - score(macro) = \frac{1}{|C|}\sum_{i=1}^{|C|}\frac{2 * precision * recall}{precision + recall} \tag{4}$$

where C is a certain class.

In all the tables further the accuracy for top-k feature selection approach is presented separately from chi-squared one. Applying χ^2 test we chose top-100, top-300, top-500, top-800, top-1000 features and used them for test. In all cases top-800 showed the better accuracy, therefore the tables contain results for this value. Tables with χ^2 approach result also contain $diff$ the difference showing how better worse the χ^2 approach showed itself in comparison to the top-k one.

In Tables 2 and 3 results for POS N-gram model are presented. We tested this model with different number of features: 1200, 1000, 800, 500, 300, 100. The best results were obtained on 1000 features therefore we focus on these results.

Table 2. Classification results for POS N-gram model. Top-k $TF - IDF$ approach

Algorithm	Avg accuracy	Avg F1-score (macro)
Support Vector Machine	**0.71**	0.69
Naive Bayes	0.63	0.61
Logistic Regression	**0.69**	0.67
K-nearest Neighbors	0.37	0.36
Decision Tree	0.40	0.39
Random Forest	0.63	0.6
Multilayer Perceptron	**0.67**	0.66
Gradient Boosting Classifier	0.51	0.50

Table 3. Classification results for POS N-gram model. Top-k χ^2 approach

Algorithm	Avg accuracy	Avg F1-score (macro)	$diff$
Support Vector Machine	**0.68**	0.67	−0.03
Naive Bayes	0.63	0.61	0.0
Logistic Regression	**0.65**	0.64	−0.04
K-nearest Neighbors	0.36	0.37	−0.01
Decision Tree	0.42	0.40	+0.02
Random Forest	0.63	0.59	0.0
Multilayer Perceptron	**0.66**	0.64	−0.01
Gradient Boosting Classifier	0.50	0.49	−0.01

In Tables 4 and 5 results for Word N-gram model are presented. For this model we used the following sizes of feature vectors: 1200, 1000, 800, 500, 300, 100. The best results were obtained on 1000-lengthed feature vectors, but for the most of algorithms there was very minor accuracy difference starting from 800 features in feature vectors.

Table 4. Classification results for Word N-gram model. Top-k $TF - IDF$ approach

Algorithm	Avg accuracy	Avg F1-score (macro)
Support Vector Machine	**0.70**	0.69
Naive Bayes	0.64	0.64
Logistic Regression	**0.65**	0.65
K-nearest Neighbors	0.17	0.14
Decision Tree	0.40	0.38
Random Forest	**0.66**	0.64
Multilayer Perceptron	0.65	0.64
Gradient Boosting Classifier	0.49	0.47

Table 5. Classification results for Word N-gram model. Top-k χ^2 approach

Algorithm	Avg accuracy	Avg F1-score (macro)	$diff$
Support Vector Machine	**0.69**	0.68	−0.01
Naive Bayes	0.62	0.60	−0.02
Logistic Regression	**0.63**	0.63	−0.2
K-nearest Neighbors	0.17	0.15	0.0
Decision Tree	0.42	0.40	+0.02
Random Forest	**0.65**	0.65	−0.01
Multilayer Perceptron	0.63	0.62	−0.02
Gradient Boosting Classifier	0.51	0.50	+0.02

In Tables 6 results for Combine model are presented. We carried out experiments to approximate the optimal ratio between number of POS N-gram and number of Word N-gram. We tested following ratio values of POS/Word N-gram: 200/100; 300/100, 400/100; 500/100; 300/50; 400/50; 500/50; 100/400; 200/400; 300/400; 300/300; 400/500. The highest accuracy was observed for ratio 300/400.

Table 6. Classification results for Combine N-gram model. Top-k $TF-IDF$ approach

Algorithm	Avg accuracy	Avg F1-score (macro)
Support Vector Machine	**0.81**	0.79
Naive Bayes	0.74	0.73
Logistic Regression	**0.79**	0.79
K-nearest Neighbors	0.55	0.55
Decision Tree	0.46	0.46
Random Forest	0.70	0.68
Multilayer Perceptron	**0.80**	0.79
Gradient Boosting Classifier	0.66	0.65

In Table 7 results of combine model with ratio 300/400 per author are presented. It might be seen that there are authors whose texts were identified with high accuracy by all algorithms and vise versa there are authors whose texts were recognized worse.

We applied χ^2 method to combine model in two different ways. The first one implied χ^2 reduction of united vectors of POS and Word N-grams (see Table 8). The idea of the second one was to reduce dimension of POS N-gram and Word N-gram vectors and then join them (see Table 9).

Below we summarize the main insights that can be carried out of conducted experiments results.

– On algorithm selection
As it can be seen from the Tables 2, 3, 4, 5, 6 and 8 the most accurate results were achieved on SVM, Multilayer Perceptron and Logistic Regression algorithms. It should be noted that two of those algorithms were stably accurate when the length of the feature vector varied (being more than 800 in general). Those algorithms were SVM and Logistic Regression. We explain it by the fact that those algorithms are ignorant to correlated features. It is interesting that Logistic Regression and SVM were still accurate enough on short feature vectors (about 300 elements) when Combine N-Gram model was applied. Such algorithms as K-nearest Neighbors and Decision Tree showed extremely poor results. It proves that K-nearest Neighbors algorithm performs poorly on high-dimensional data as it is not resistant to noise, while Decision Tree algorithm is hardly applicable to classification task where continuous feature space is implied.

- On feature selection approach
  Having compared two types of feature selection approaches we could state that chi-squared approach was less beneficial in most cases. Rather simple approach of ranking the features by $TF-IDF$ measure showed slightly lower accuracy.

Table 7. Classification results per author

Author	SVM	NB	LG	MLP	RF	kNN	GBC	DT
a1	36	20	34	33	35	35	30	24
a2	29	26	27	25	25	8	23	12
a3	34	25	35	34	23	17	20	12
a4	30	31	29	28	28	23	23	16
a5	31	27	29	30	26	11	25	24
a6	36	33	35	35	33	22	25	20
a7	27	26	29	29	22	18	27	17
a8	35	34	34	34	35	21	29	18
a9	38	39	38	38	38	19	25	26
a10	23	24	23	23	11	17	19	6
a11	19	20	18	23	11	16	15	9
a12	33	33	32	32	30	25	29	13
a13	36	38	36	35	40	16	27	28
a14	29	22	30	28	16	14	24	8
a15	37	29	35	37	33	25	29	27
a16	36	37	36	36	35	35	34	20
a17	40	39	38	40	38	27	26	15

Table 8. Classification results for Combine N-gram model. Top-k χ^2 approach (1)

Algorithm	Avg accuracy	Avg F1-score (macro)	$diff$
Support Vector Machine	**0.78**	0.77	−0.03
Naive Bayes	0.73	0.71	−0.01
Logistic Regression	**0.77**	0.75	−0.02
K-nearest Neighbors	0.30	0.30	−0.25
Decision Tree	0.47	0.46	−0.01
Random Forest	0.72	0.70	−0.02
Multilayer Perceptron	**0.77**	0.76	−0.03
Gradient Boosting Classifier	0.63	0.63	−0.03

Table 9. Classification results for Combine N-gram model. Top-k χ^2 approach (2)

Algorithm	Avg accuracy	Avg F1-score (macro)	$diffacc$
Support Vector Machine	**0.80**	0.76	−0.01
Naive Bayes	0.74	0.73	+0.00
Logistic Regression	**0.76**	0.75	−0.03
K-nearest Neighbors	0.37	0.36	−0.18
Decision Tree	0.44	0.43	−0.02
Random Forest	0.73	0.72	+0.03
Multilayer Perceptron	**0.79**	0.79	−0.01
Gradient Boosting Classifier	0.63	0.63	+0.03

– On feature extraction approach
  We got a significant improvement in accuracy of classification when combining two POS and Word N-gram models especially in the cases when word N-gram number was superior to number of POS N-grams. In addition, due to the reduction of dimension of feature vectors from 1000 to 500, the speed of the classification has increased.

5 Conclusion

In this paper we investigated the problem of Author Identification for Russian web-posts. The purpose of this study was to find more appropriate classification method from class of Machine Learning algorithms. The performance of applied method was evaluated during set of experiments on data set which contained 40 texts per each of 17 authors. We tested three feature extraction models such as POS N-grams, Word N-grams and combination of ones. For estimation of the performance 10-cross-validation approach was used. The highest accuracy was achieved on Support Vector Machine with linear kernel, Multilayer Perceptron and Logistic Regression. These algorithms showed the best results for all observed models on short Russian texts. The empirical results of our experiments encourage using Combine model for text representation and applying $TF - IDF$ ranking for feature selection in such an environment. As experiments showed this combination of approaches might significantly increase the classification accuracy.

References

1. Fissette, M.: Author identification in short texts (2010)
2. Ganesh, H.B.B., Reshma, U., Kumar, M.A.: Author identification based on word distribution in word space. In: 2015 International Conference on Advances in Computing, Communications and Informatics (ICACCI), pp. 1519–1523, August 2015. https://doi.org/10.1109/ICACCI.2015.7275828

3. Howedi, F., Mohd, M.: Text classification for authorship attribution using Naive Bayes classifier with limited training data. In: Computer Engineering and Intelligent Systems (2014)
4. Jenkins, J., Nick, W., Roy, K., Esterline, A.C., Bloch, J.C.: Author identification using sequential minimal optimization. In: SoutheastCon 2016, pp. 1–2 (2016)
5. Kanhirangat, V., Gupta, D.: Text plagiarism classification using syntax based linguistic features. Expert Syst. Appl. **88**, 448–464 (2017). https://doi.org/10.1016/j.eswa.2017.07.006. http://www.sciencedirect.com/science/article/pii/S09574174475X1730
6. Kapočiūtė-Dzikienė, J., Venčkauskas, A., Damaševičius, R.: A comparison of authorship attribution approaches applied on the Lithuanian language. In: 2017 Federated Conference on Computer Science and Information Systems (FedCSIS), pp. 347–351, September 2017. https://doi.org/10.15439/2017F110
7. Khonji, M., Iraqi, Y., Jones, A.: An evaluation of authorship attribution using random forests. In: 2015 International Conference on Information and Communication Technology Research (ICTRC), pp. 68–71, May 2015. https://doi.org/10.1109/ICTRC.2015.7156423
8. Korobov, M.: Morphological analyzer and generator for Russian and Ukrainian languages. In: Khachay, M.Y., Konstantinova, N., Panchenko, A., Ignatov, D.I., Labunets, V.G. (eds.) AIST 2015. CCIS, vol. 542, pp. 320–332. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-26123-2_31
9. Largeron, C., Juganaru-Mathieu, M., Frery, J.: Author identification by automatic learning. In: IEEE International Conference on Document Analysis and Recognition (ICDAR 2015), Nancy, France, August 2015. https://hal.archives-ouvertes.fr/hal-01223252
10. Meina, M., et al.: Ensemble-based classification for author profiling using various features notebook for pan at CLEF 2013. In: CLEF (2013)
11. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
12. Pokou, Y.J.M., Fournier-Viger, P., Moghrabi, C.: Authorship attribution using variable length part-of-speech patterns. In: ICAART (2016)
13. Pranckevičius, T., Marcinkevičius, V.: Application of logistic regression with part-of-the-speech tagging for multi-class text classification. In: 2016 IEEE 4th Workshop on Advances in Information, Electronic and Electrical Engineering (AIEEE), pp. 1–5, November 2016. https://doi.org/10.1109/AIEEE.2016.7821805
14. Reddy, T.R., Vardhan, B.V., Reddy, P.V.: N-gram approach for gender prediction. In: 2017 IEEE 7th International Advance Computing Conference (IACC), pp. 860–865 (2017)
15. Rogati, M., Yang, Y.: High-performing feature selection for text classification. In: Proceedings of the Eleventh International Conference on Information and Knowledge Management, CIKM 2002, pp. 659–661. ACM, New York (2002). https://doi.org/10.1145/584792.584911
16. Stamatatos, E.: A survey of modern authorship attribution methods. J. Am. Soc. Inf. Sci. Technol. **60**(3), 538–556 (2009). https://doi.org/10.1002/asi.v60:3
17. Vorobeva, A.A.: Examining the performance of classification algorithms for imbalanced data sets in web author identification. In: 2016 18th Conference of Open Innovations Association and Seminar on Information Security and Protection of Information Technology (FRUCT-ISPIT), pp. 385–390, April 2016. https://doi.org/10.1109/FRUCT-ISPIT.2016.7561554

Implementing a Face Recognition System for Media Companies

Arturs Sprogis[1](✉), Karlis Freivalds[1], and Elita Cirule[2]

[1] Institute of Mathematics and Computer Science, University of Latvia,
Raina blvd. 29, Riga 1459, Latvia
{arturs.sprogis,karlis.freivalds}@lumii.lv
[2] LETA, Marijas 2, Riga 1050, Latvia
elita.cirule@leta.lv

Abstract. During the past few years face recognition technologies have greatly benefited from the huge progress in machine learning and now have achieved precision rates that are even comparable with humans. This allows us to apply face recognition technologies more effectively for a number of practical problems in various businesses like media monitoring, security, advertising, entertainment that we previously were not able to do due to low precision rates of existing face recognition technologies. In this paper we discuss how to build a face recognition system for media companies and share our experience gained from implementing one for Latvian national news agency LETA. Our contribution is: which technologies to use, how to build a practical training dataset, how large should it be, how to deal with unknown persons.

Keywords: Face recognition · Media companies · System implementation

1 Introduction

A face recognition problem has been researched for decades and for a long time hand-crafted or machine assisted feature detection techniques were the dominant. The latest such face recognition techniques [1–3] use sophisticated algorithms to select up to tens of thousands of parameters to represent a face. But they achieve high accuracy only in constrained environments where faces are frontally positioned, and a variety of conditions like lighting, expression, and occlusion are restricted. Obviously in real life images these conditions are rarely satisfied, and, therefore, in unconstrained environment the accuracy rates are significantly lower.

In contrast to hand-crafted techniques, machine learning approaches let a computer (instead of human) figuring out which face parameters are important to measure. Since these parameters are extracted from the training data, the recognition accuracy depend more on the quality and variety of the training data and less on the environment constraints. In addition, machine learning models typically use 100 to 2000 dimensions to represent a face, in contrast to tens of thousands used by hand-crafted methods. As a result, machine learning techniques are more effective for representing and processing

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 328–341, 2018.
https://doi.org/10.1007/978-3-319-97571-9_26

faces, as well as they are more flexible and accurate to recognize people in real life images with no constraining conditions.

In this paper we are discussing how to apply the latest scientific and technological achievements in face recognition from specialized face recognition companies as well as tech giants like Google, Facebook, Baidu to implement a face recognition system for media companies utilizing state-of-the-art face recognition machine learning models. Typically, media companies own large archives of images and videos in which persons have to be tagged. Accomplishing this task requires executing a number of sub-tasks like extracting training datasets of sample images with a large number of identities (possibly thousands), additionally recognizing unknown people (that is, those who do not belong to our training dataset), selecting an appropriate algorithm for each step in the face recognition pipeline.

The structure of the paper is the following. In Sect. 2 we will explain the pipeline of modern face recognition technology. In Sect. 3 we will discuss how to put together all the building blocks to implement the actual face recognition system.

2 Face Recognition Pipeline

A typical modern face recognition pipeline consists of four steps: detection, alignment, representation and classification [4].

2.1 Detection and Alignment

The first step in the face recognition pipeline is to detect faces. To detect faces we have to locate face areas in images. As a result a list of locations of faces is produced in this step. An example of face detection is demonstrated in Fig. 1.

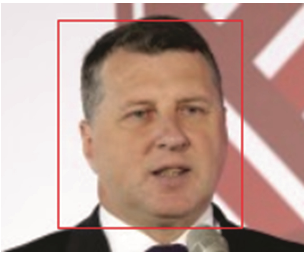


Fig. 1. An example of face detection

When we have detected faces, they most probably are turned in different directions. Obviously such faces look totally different to a computer. To account for this, faces are aligned so that the eyes and lips are always centered. This significantly simplifies the face recognition step for the computer.

2.2 Representation

In this step, an n-dimensional vector (also called "embedding" vector) is computed to represent each face. These vectors have a characteristic that vectors representing one and the same person are geometrically close in n-dimensional space, whereas vectors representing different persons are geometrically farther from each other. Traditionally, the L2 or cosine distance is used to measure the distance between vectors, and then some experimental threshold is determined to distinguish whether two vectors represent one and the same person or two different persons. An example of face recognition is demonstrated in Fig. 2.

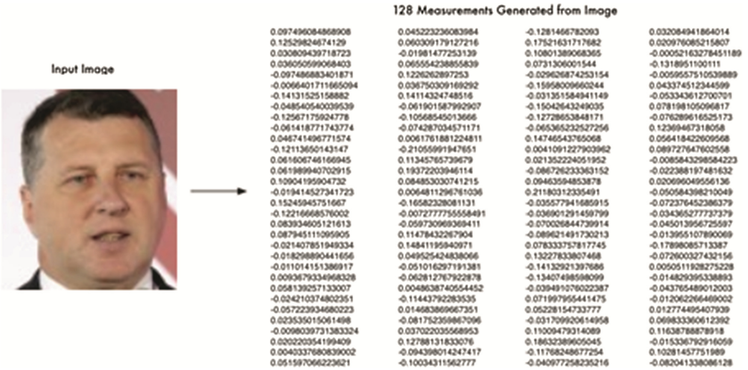


Fig. 2. An example of face representation as a 128 dimensional vector

The face representation step is the most important and complex of all steps in the face recognition pipeline because the overall accuracy rates mostly depend on the way we compute face representations. Currently, the best results are demonstrated by face recognition systems that perform the following two steps. Firstly, they transform faces to n-dimension vector using deep convolutional neural network (CNN) [5]. Secondly, they perform dimensionality reduction, if necessary. Commonly used methods for this task are PCA [6], Joint Bayesian [7] and Metric learning [8].

Next we will review some of the most accurate approaches for face representation computing which are selected by their performance on the popular LFW [9] dataset.

MMDFR. MMDFR [10] is a solution that achieves 99.02% accuracy rate on LFW dataset. In the first step the system aligns faces to 230 × 230 pixels and then transforms them to 3D model. Then the 3D model is cropped and passed to 8 different CNNs. The representation vector is computed by combining 8 vectors by applying Stacked Auto-Encoder method for dimensionality reduction. The system is trained on a data set containing more than 9,000 identities.

Face++. FACE++ [11] is a solution that achieves 99.50% accuracy rate on LFW dataset. The system is trained on a data set consisting of 500,000 images with 10,000 identities. The model consists of 4 different models each recognizing a certain face area. The representation vector is computed by combining 4 vectors by applying PCA method.

DeepID3. DeepID3 [12] is a solution that achieves 99.53% accuracy rate on LFW dataset. The system consists of two CNN models VGG net [13] and GoogLeNet [14] that are extended with Supervisory signals method [15]. For one model the original face is passed, for the second model the horizontally rotated faces. The representation vector is computed by combining 2 vectors in a single vector having 300,000 dimensions. By applying PCA method the dimensionality is reduced to 300. The system is trained on a data set consisting of 300,000 images.

Facenet. Facenet [16] is a solution that achieves 99.63% accuracy rate on LFW dataset. The system is implemented by Google. The model consists of 1 CNN with 140 million parameters and is trained on a data set consisting of hundreds of millions of images. The resulting representation vector has 128 dimensions.

Daream. Daream [17] is a commercial solution that achieves 99.68% accuracy rate on LFW dataset. The publicly available information tells that the system is trained on a data set consisting of 1.2 million images with 30,000 identities. The final model consists of 4 different models - Residual network [18], Wide-residual network [19], Highway path network [20] and Alexnet [21]. The representation vector is computed by combining 4 vectors by applying Joint Bayesian method for dimensionality reduction.

Baidu. Baidu [22] is a commercial solution that achieves 99.77% accuracy rate on LFW dataset. The publicly available information tells that the system is trained on a data set consisting of 1.2 million images with 18,000 identities. The final model consists of 9 different models each recognizing a certain face area. The representation vector is computed by combining 9 vectors by applying learning with triplet loss method to obtain a single 128-dimensional vector.

Dahua-FaceImage. Dahua-FaceImage [23] is a commercial solution that achieves 99.78% accuracy rate on LFW dataset. The publicly available information tells that the system is trained on a data set consisting of 2 million images with 20,000 identities. The final model consists of 30 different CNNs, and the representation vector is computed as a combination of 30 vectors by applying *Joint Bayesian* method for dimensionality reduction.

3 Implementation

So far we have seen an overview of a general face recognition pipeline and have reviewed various techniques to compute face representations. In this section we will discuss how to implement the actual face recognition system by applying these techniques.

3.1 Model

To compute face representations we must have a trained model. One option is to select one of the previously described approaches to train the model by ourselves, or another is to use the already trained model. To train a model from scratch we need a large dataset with labeled images as well as computing resources. On the other hand, there are available already trained models, for instance, [24, 25]. These models are trained on publicly available datasets CASIA-WebFace [26], FaceScrub [27] and MS-Celeb-1M [28]. Thus, if we have enough computing resources and our data set is larger than publicly available data sets, then it is reasonable to train a model by ourselves. Otherwise, we would recommend selecting an existing solution. In general, they have very decent precision rates. For [24] the accuracy rate is 97.3%, for [25] it is 99.2% on LFW datasets in comparison to 99.63% and 99.78% for solutions from Google and Baidu.

3.2 Classification

When we have computed face representations, we have to decide how to classify them, or in other words, how to attach the most appropriate name. In general, for many classification tasks SVM [29] algorithm is a popular choice. However, it does not fit well for our face classification task when there are known and unknown persons. To classify unknown persons we need not only the identity of the classified person but also the confidence score to determine the likelihood level of the classified person to be able to decide between the known and the unknown persons. Although SVM returns both, the identity and confidence score, the problem is that the confidence score is computed as a probability depending on the number of identities in the training dataset, and therefore it is not possible to have one particular threshold value because its values varies as a number of identities change (as a set of identities grows, the confidence score decreases).

We are suggesting to use an alternative approach: K-nearest neighbors (KNN) algorithm. In particular, the algorithm computes the k nearest points and the corresponding spatial distances for the given point. In context of face recognition, the algorithm for the given face selects the k most similar faces from the training dataset and their corresponding distances. Then we have to decide how to classify the known and the unknown persons. The algorithm we used is the following. First, we pick k most similar candidate faces (we used k = 12, but regarding the second step whether k is selected sufficiently large, it has minimal effect on the accuracy rate). Second, from the top k candidates we select those with distances smaller than some given distance threshold. This will give us the top similar faces, and then we have to decide whether it is a known person and give its name, or declare it as unknown. To achieve this we use a voting mechanism that attaches the identity having the majority of the votes or hits some threshold. To explain it in a greater detail, we will assume that after the second step we have selected the 10 top similar faces and the voting threshold is 4. Now we may have multiple cases. One case is that the majority of the faces belong to the same identity X (for instance, 6 of 10), and this number is greater than the voting threshold (6 is greater than 4), then we declare that the given face belongs to X. Another case is when there is not a majority for one particular identity, for instance, 4 faces belong to X and Y, and the rest belong to Z. In that case we measure average distances for X and Y additionally to decide the identity. Finally, we may have a case when none of the identities hits the voting threshold, and in that case we declare the given face as unknown. For instance, if 2 faces belong to the five different identities, then none of them satisfy the voting threshold restriction (2 is not greater than 4).

Nevertheless, in practice it is wise to add some categories during the classification. For example, in LETA project we introduced four categories – very high (distance threshold minus 4x bias), high (distance threshold minus 3x bias), medium (distance threshold minus 2x bias) and low (distance threshold minus bias) where bias is equal 0.06.

3.3 Training Dataset

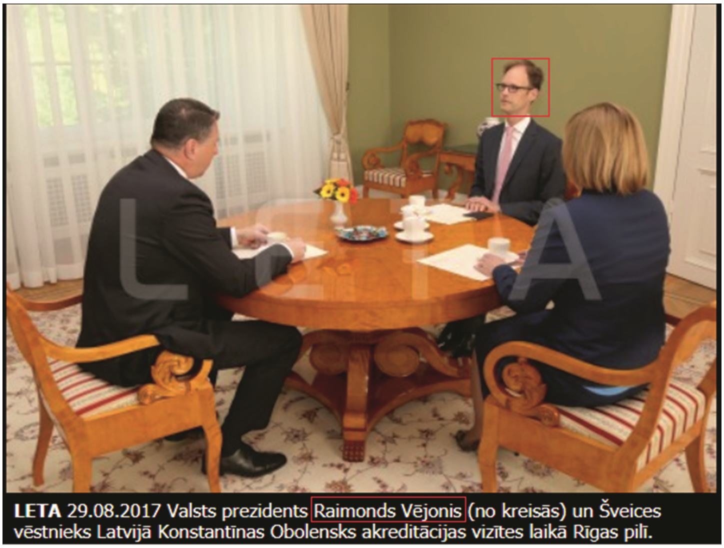
Until now we have discussed technologies handling the engineering complexity, but one of the key problems to successfully apply face recognition technologies is data. Thus before we can recognize someone in a picture or video, we have to build a training dataset containing some number of images per each person we want to recognize. To achieve applicable results we need at least 6 sample images per identity, however, the optimal number would be 25–45 (see Sect. 3.4). The rule of thumb is that the more samples with different variety we have, the more accurate results we will get. This is especially important to cover real life situations when faces are in different facial expressions, various poses, image resolutions, lighting, etc.

Thus to build a system being able to recognize thousands of identities, we need to collect tens or even hundreds of thousands of training images accordingly. Performing such a task manually is slow, labor-intensive and expensive, and therefore not scalable. While examining alternatives, we noticed that most of the images from LETA archive have an additional description explaining the location, event and the persons in the image (see Fig. 3) which we found very useful to provide an automated support for identity extraction.

The idea for the identity extraction algorithm is straightforward. First, we perform a face detection and check if there is exactly one face in the image. Then we perform a description analysis by applying advanced natural language processing algorithms [30] to extract person entities (it has to be noted that in Latvian language word endings change depending on context that makes an entity extraction more complicated) and then we check if there is exactly one entity. Thus, if there is exactly one face detected and there is exactly one entity extracted, then we assume that the detected face belongs to the extracted entity.

However, we found that there are scenarios when this assumption does not hold. The problem is that the entity extracted cannot always detect foreign names and therefore instead of two names sometimes only one is extracted causing face-entity mismatch (see Fig. 3).

To deal with face-entity mismatches, we extended the identity extraction process with post-processing step. The idea of this step is to iterate over the collected training dataset and find possibly higher number of face-entity mismatches. To accomplish the task we applied our face recognition algorithm for every image in the training dataset. Since we already know to which identity the image belongs, we can easily compare if



LETA 29.08.2017 Valsts prezidents Raimonds Vējonis (no kreisās) un Šveices vēstnieks Latvijā Konstantīnas Obolensks akreditācijas vizītes laikā Rīgas pilī.

Fig. 3. An example of face-entity mismatch where Latvian president Raimonds Vejonis is incorrectly identified

the identity proposed by the face recognition algorithm is equal to the identity image belongs. Although the idea of the post-processing step is simple, it proved to be very effective since it detected that approximately 10% of images in training dataset were face-entity mismatches, thus helping to increase the quality of the training dataset.

As a result, by performing our identity extraction algorithm we managed to extract 4400 unique person names with a total of 95,000 images from LETA archive.

3.4 Experimental Evaluations

In practice we have to determine three things: how does the number of images per person affect the overall accuracy rate, what are the optimal KNN parameters and what the recognition speed is. To answer these questions we performed a number of experiments, and rated the results on a test dataset containing 1255 faces where 892 are faces of known persons of 186 identities and 363 faces are faces of unknown persons. All of the images were selected from LETA image archive.

To find an answer to the first question, we trained 50 different models having randomly selected 1 to 50 images per person in the training dataset accordingly. Figure 4 represents the obtained results, where X axis represent the number of images per person and Y axis represent the error rate on our test dataset. The error rate is computed as total correct images (known and unknown) divided by total images (892 known and 363 unknown). We can see that 6% error rate is reached when there are at least 6 images per person, 5% error rate when there are at least 24 images per person and 4.2% error rate (the lowest) when there are at least 45 images per person. Thus, the more images per person we have, the more accurate results we get.

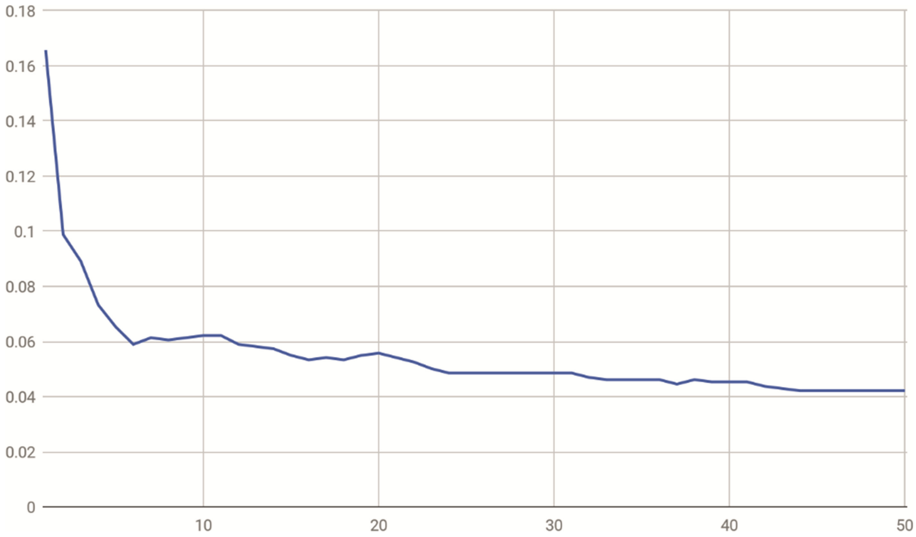


Fig. 4. An error rate depending on the images count per person

Figure 5 represents the error rate only among known persons.

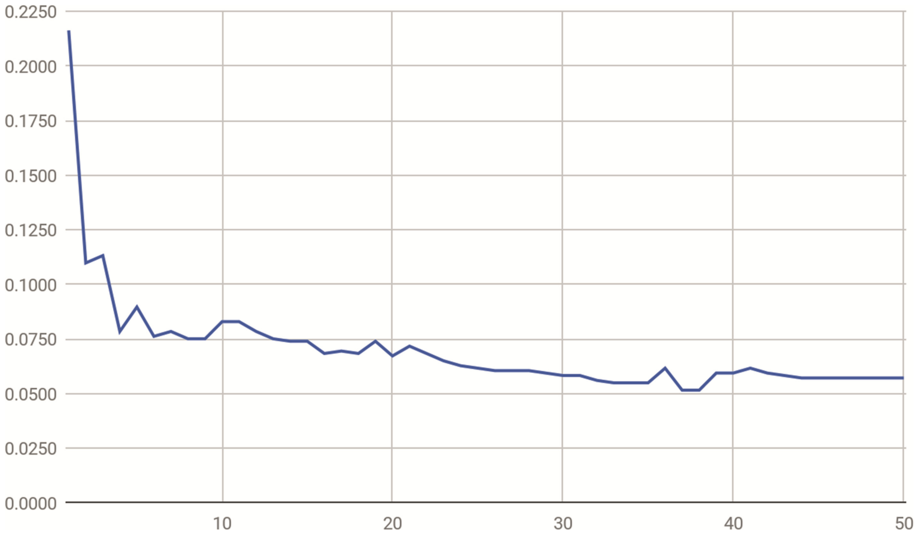


Fig. 5. An error rate of know persons depending on the images count per person

Figure 6 represents the error rate only among unknown persons.

We have also tested SVM algorithm on a dataset containing only images of known persons (see Fig. 7), and we can see that SVM results are slightly better than KNN results, still KNN is competitive, but with SVM we cannot detect unknown persons.

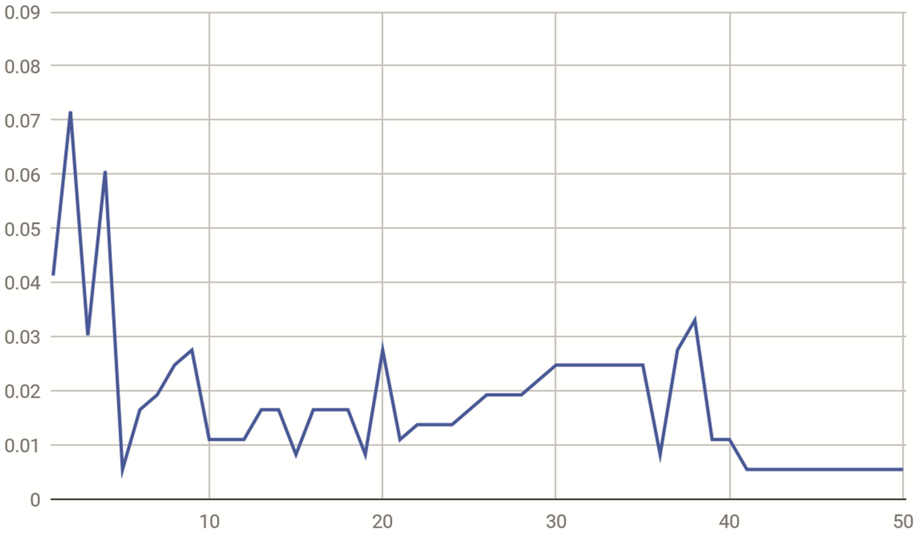


Fig. 6. An error rate of unknown persons depending on the images count per person

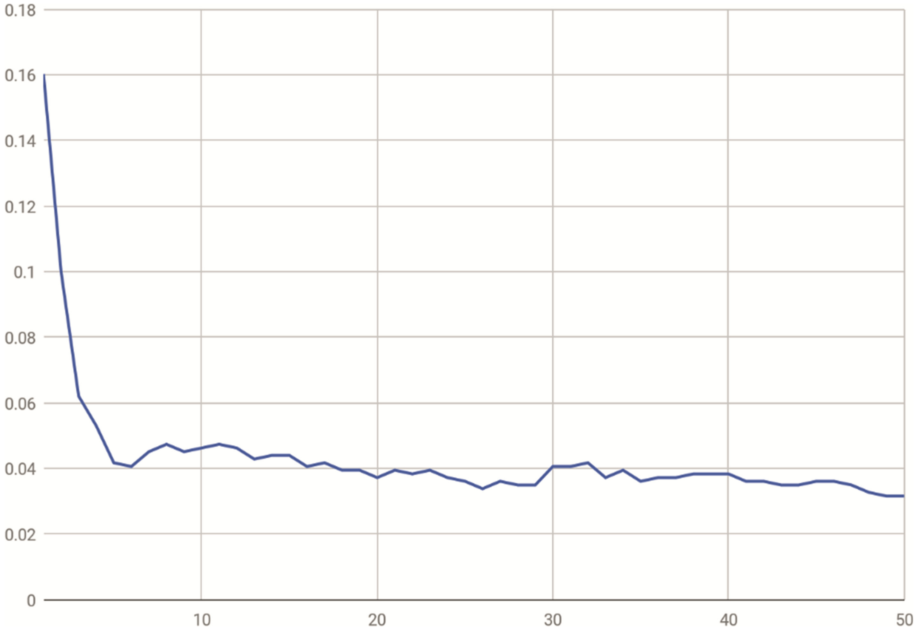


Fig. 7. An error rate of known persons depending on the images count per person using SVM algorithm

To find the optimal KNN parameters, we also performed the same 50 experiments having 1 to 50 images per person where count thresholds were from 1 to 4, and the distance thresholds in range from 0.75 to 1 with a step size 0.05. Figures 8, 9, 10 and 11 represent how the accuracy changes depending on different voting and distance thresholds. Figures show that if the training dataset contains less than 10 images, then the distance threshold have to be relatively high to achieve the best performance. Whereas, if the size of the training dataset is getting larger than 10, then the tendency is that the best performance is achieved when the distance threshold is relatively small.

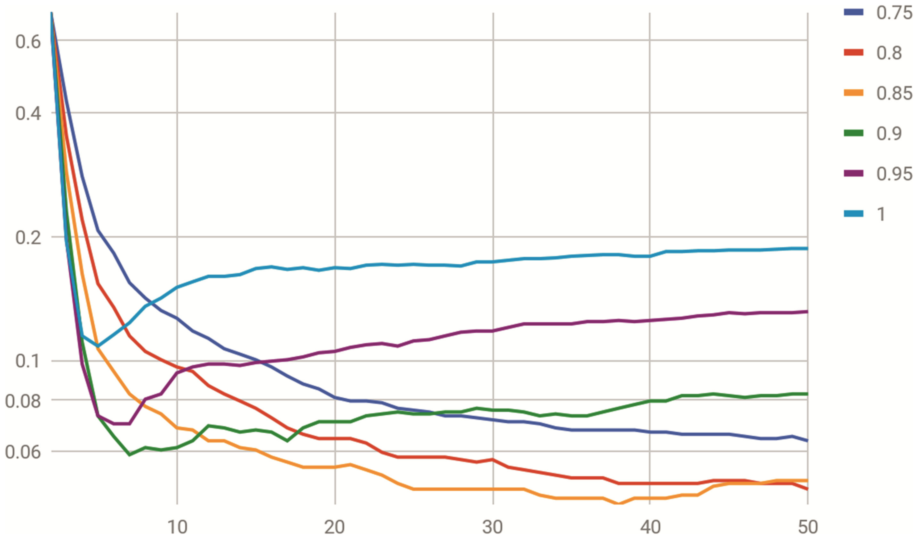


Fig. 8. Error rates with voting threshold 1 and various distance thresholds

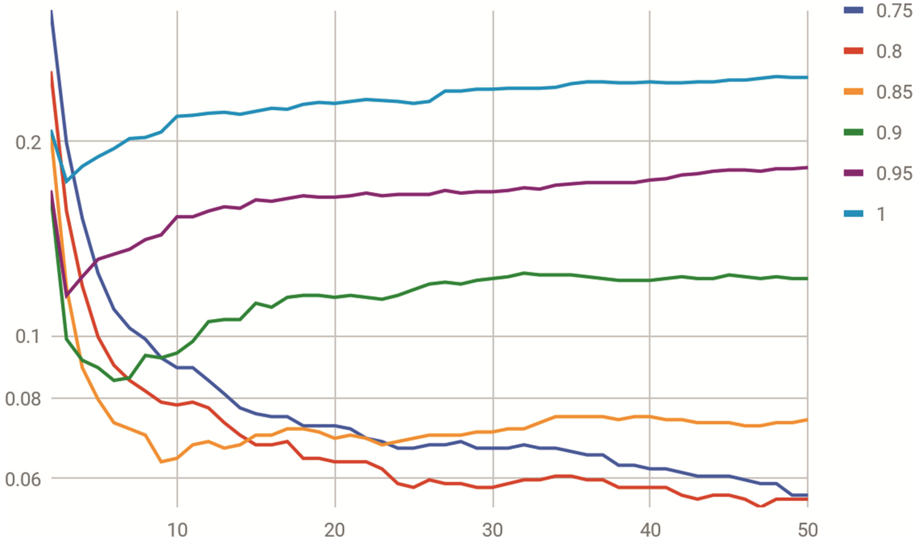


Fig. 9. Error rates with voting threshold 2 and various distance thresholds

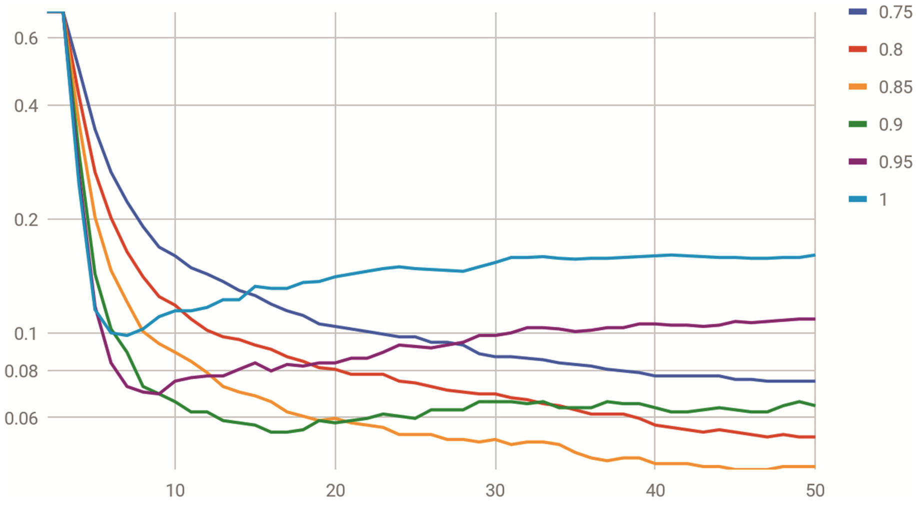


Fig. 10. Error rates with voting threshold 3 and various distance thresholds

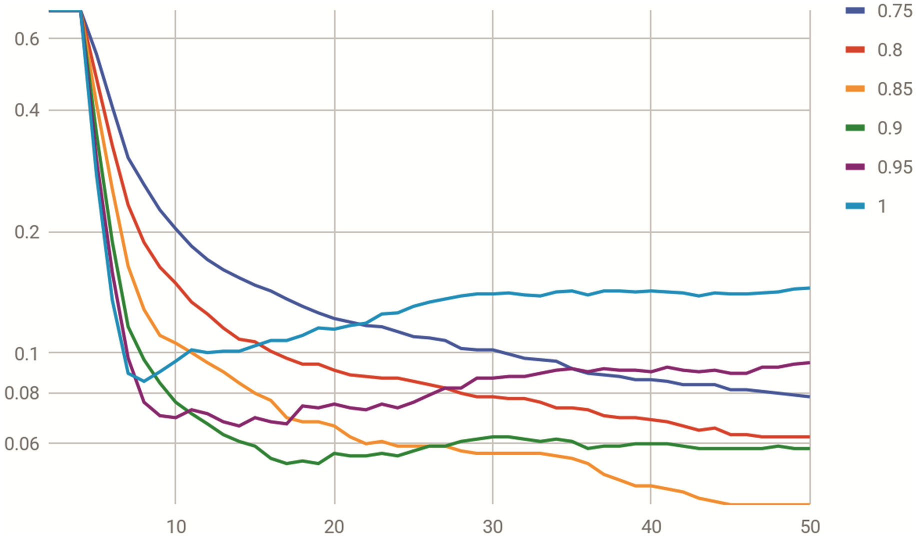


Fig. 11. Error rates with voting threshold 4 and various distance thresholds

Figure 12 represents average error rates by various distance thresholds and we can see that the smallest error is achieved having 0.85 distance threshold value.

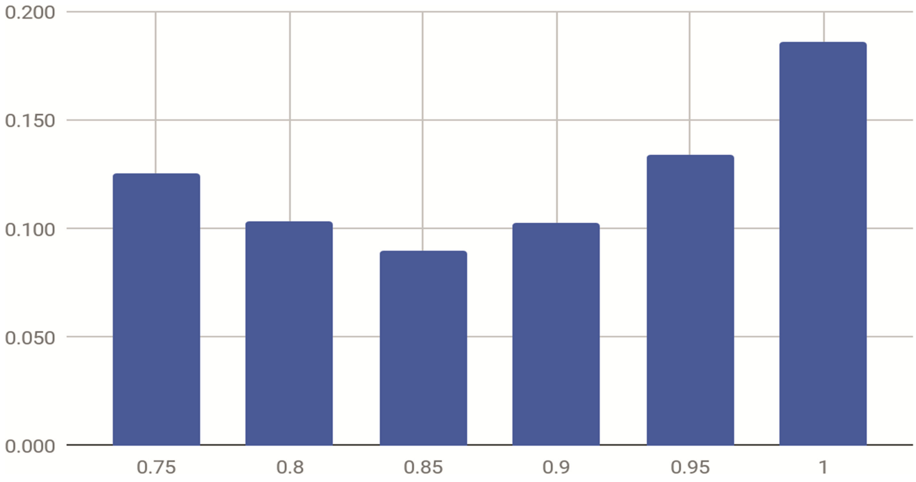


Fig. 12. Average error rates by various distance thresholds

Our experiments show that it takes on average 0.3 s to detect and recognize one face. The experiments were performed on NVIDIA GPU TITAN X 12 GB graphical processor.

4 Conclusions

In this paper we have discussed a general pipeline to implement a face recognition system for media companies and shared our experience implementing one for the Latvian national news agency LETA. To implement a system, we have to make a number of decisions, and the stack we have implemented in LETA project is as following. To compute face representation we selected the pre-trained model [25], for classification we used KNN algorithm with parameters depending on a number of images per person in the training dataset. To build a training dataset we used our custom made algorithm which automatically extracted 4400 identities with the total of 95,000 sample images from LETA image archive. We tested the implemented system on the dataset with 1255 faces where 892 were faces of known persons of 186 identities and 363 faces were of unknown persons and the accuracy rate we achieved was 95.78% when there are at least 45 images per person in the training dataset. We have also tested the algorithm on a number of YouTube videos where Latvian politicians participated, and the accuracy rate stayed the same. Thus the implemented system is applicable on different datasets as well.

While implementing the system we performed experimental evaluations regarding optimal training dataset size and optimal classification algorithm parameters. Experiments show that we have to build a training dataset of images that contain at least 6 images per person but optimally they are 25–45 images per person, whereas KNN parameters have to be adjusted according to the image count per person in the training dataset.

Acknowledgement. The research leading to these results has received funding from the research project "Competence Centre of Information and Communication Technologies" of the EU Structural funds, contract No. 1.2.1.1/16/A/007 signed between IT Competence Centre and Central Finance and Contracting Agency, Research No. 2.5 "Automated visual material recognition and annotation system for LETA archive".

References

1. Cao, X., Wipf, D., Wen, F., Duan, G., Sun, J.: A practical transfer learning algorithm for face verification. In: Proceedings of ICCV (2013)
2. Barkan, O., Weill, J., Wolf, L., Aronowitz, H.: Fast high dimensional vector multiplication face recognition. In: Proceedings of ICCV (2013)
3. Phillips, P.J., et al.: An introduction to the good, the bad, & the ugly face recognition challenge problem. In: FG (2011)
4. Taigman, Y., Yang, M., Ranzato, M.A., Wolf, L.: DeepFace: closing the gap to human-level performance in face verification. In: Computer Vision and Pattern Recognition (2014)
5. LeCun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W., Jackel, L.D.: Backpropagation applied to handwritten zip code recognition. Neural Comput. **1**(4), 541–551 (1989)
6. Joliffe, I.T.: Principal Component Analysis. Springer, New York (2002). https://doi.org/10.1007/b98835
7. Chen, D., Cao, X., Wang, L., Wen, F., Sun, J.: Bayesian face revisited: a joint formulation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7574, pp. 566–579. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33712-3_41
8. Xing, E.P., Ng, A.Y., Jordan, M., Russell, S.: Distance metric learning with application to clustering with side-information. In: Proceedings of the 15th Advances in Neural Information Processing Systems (NIPS 2002), pp. 521–528 (2002)
9. Huang, G.B., Learned-Miller, E.: Labeled faces in the wild: updates and new reporting procedures. University of Massachusetts, Amherst, Technical report UM-CS-2014-003 (2014)
10. Ding, C., Tao, D.: Robust face recognition via multimodal deep face representation. IEEE Trans. Multimed. **17**(11), 2049–2058 (2015)
11. FACE++. https://arxiv.org/pdf/1501.04690v1.pdf
12. Sun, Y., Liang, D., Wang, X., Tang, X.: DeepID3. face recognition with very deep neural networks (2014)
13. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition (2014). arXiv preprint: arXiv:1409.1556
14. Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V., Rabinovich, A.: Going deeper with convolutions (2014). arXiv:1409.4842 [cs.CV]
15. Sun, Y., Wang, X., Tang, X.: Hybrid deep learning for face verification. In: Proceedings of ICCV (2013)
16. Schroff, F., Kalenichenko, D., Philbin, J.: FaceNet: a unified embedding for face recognition and clustering (2015). arXiv:1503.03832 [cs.CV]
17. Daream. http://www.daream.com/
18. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38

19. Zagoruyko, S., Komodakis, N.: Wide Residual Networks (2016). arXiv:1605.07146 [cs.CV]
20. Srivastava, R.K., Greff, K., Schmidhuber, J.: Deep Learning Workshop (ICML 2015) (2015)
21. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, vol. 1, pp. 1097–1105 (2012)
22. Liu, J., Deng, Y., Bai, T., Wei, Z., Huang, C.: Targeting ultimate accuracy: Face recognition via deep embedding (2015). arXiv:1506.07310v4 [cs.CV]
23. Dahua-FaceImage. http://www.dahuatech.com/recognition/index.php
24. OpenFace. http://cmusatyalab.github.io/openface/
25. Face recognition using Tensorflow. https://github.com/davidsandberg/facenet
26. CASIA-WebFace. http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html
27. FaceScrub. http://vintage.winklerbros.net/facescrub.html
28. MS-Celeb-1M. https://www.microsoft.com/en-us/research/project/ms-celeb-1m-challenge-recognizing-one-million-celebrities-real-world/
29. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
30. Paikens, P.: Latvian newswire information extraction system and entity knowledge base. In: Human Language Technologies – The Baltic Perspective. Frontiers in Artificial Intelligence and Applications, vol. 268, pp. 119–125. IOS Press (2014)

Applications and Case Studies

What Language Do Stocks Speak?

Marko Poženel and Dejan Lavbič(✉)

Faculty of Computer and Information Science, University of Ljubljana,
Večna pot 113, 1000 Ljubljana, Slovenia
{Marko.Pozenel,Dejan.Lavbic}@fri.uni-lj.si

Abstract. Stock prediction is a challenging and chaotic research area where many variables are included with their effects being complex to determine. Nevertheless, stock value prediction is still very appealing for researchers and investors since it might be profitable, yet the number of published research papers remains to be relatively small. The employment of advanced data analysis techniques has already been suggested by previous researches, such as the use of neural networks for stock price prediction, but practical implications of the majority of approaches are limited as they are concerned mainly with a prediction accuracy and less with the success in real trading with consideration of trading fees. We propose a novel approach for stock trend prediction that combines Japanese candlesticks (OHLC trading data) and neural network based group of models Word2Vec. Word2Vec is usually utilized to produce word embeddings in natural language processing tasks, while we adopt it for acquiring semantic context of words in candlesticks' sequence, where clustered candlesticks represent stock's words. The approach is employed for the extraction of useful information from large sets of OHLC trading data to improve prediction accuracy. In evaluation of our approach we define a trading strategy and compare our approach with other popular prediction models – Buy & Hold, MA and MACD. The evaluation results on Russell Top 50 index are encouraging – the proposed Word2Vec approach outperformed all compared models on a test set with a statistical significance.

Keywords: Stock price prediction · Trading strategy · Word2Vec
NLP

1 Introduction

Forecasting trends and the future value of stocks has always been an interesting topic for both investors and research community. However, the number of successful researches and published papers is still very low. The reason is simple, usually nobody wants to publish an algorithm that solves one of the issues that might be most profitable. Nonetheless, there are many approaches to forecasting the future stock values, where the most influential are: (i) technical analysis [23] and (ii) fundamental analysis [1].

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 345–359, 2018.
https://doi.org/10.1007/978-3-319-97571-9_27

Fundamental analysis of financial markets involves detailed analysis of the company's business, various news about the enterprise and the prediction of future growth. It deals with linking current company's financial data to future earnings and evaluation of how it will affect company's value. A large number of factors have to be included in the evaluation [1]. Several approaches that attempt to automate stock trading based on processing of unstructured text sources such as news articles, company reports or individual posts [4,16,22], are typically based on Natural Language Processing Algorithms (NPA).

The second approach to trading is based solely on the basis of historical price changes and technical analysis. Technical analysis provides data for trading decisions largely on the basis of visual inspection of past trend movements, without employing any part of fundamental analysis [23]. Proponents of technical analysis claim that all necessary information for forecasting the stock price trends are already included in the stock price itself. They point out that events in the history are repeated and that stock prices can be forecasted based on current trends [18].

Very popular technical method to convey the growth and decline of the demand and supply in financial markets is the candlestick trading strategy [9,17]. It is one of the oldest technical analyses techniques with origins in 18th century where it was used by Munehisa Homma for trading with rice. He analysed rice prices back in time and acquired huge insights to the rice trading characteristics. Japanese candlestick charting technique is a primary tool to visualize the changes in a commodity price in a certain time span. Nowadays candlestick charting technique can be found in almost every software and on-line charting packages [5]. Although the researchers are not in complete agreement about its efficiency, many researchers are investigating its potential use in various fields [5,6,8,10,18]. To visualize Japanese candlestick at a certain time grain (e.g. day, hour), four key data components of a price are required: starting price, highest price, lowest price and closing price. This tuple is called OHLC (Open, High, Low, Close). Figure 1 shows an example of a Japanese white and black body candlestick with the notation used in this paper. When the candlestick body is filled, the closing price of the session was lower than the opening price. If the body is empty, the closing price was higher than the opening price. The thin lines above and below the rectangle body are called shadows and represent session's price extremes. There are many types of Japanese candlesticks with their distinctive names. Figure 2 shows only some of the most commonly seen in candlestick charts. Each candlestick holds information on trading session and becomes even more important, when it is an integral part of certain sequence.

The work presented in this paper is an attempt to create a simplified OHLC language (i.e. simplified language of Japanese candlesticks from OHLC data) that is later used as an input for Word2Vec algorithm [13] that can learn the vector representations of words in the high-dimensional vector space. We believe that it is possible to learn rules and patterns using Word2Vec and use this knowledge to predict future trends in stock value. Despite many developed models and predictive techniques, measuring performance of the stock prediction models can present a challenge. For example, Jasemi et al. [5] used hit ratio to evaluate the performance of the models but neglected financial success of a model. Therefore, one of the research goals of this paper is also to utilize a simple method for testing the performance of forecasting models, the result of which is the financial success or yield of the tested model.

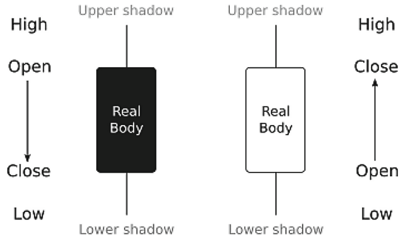


Fig. 1. The presentation of Japanese candlestick.

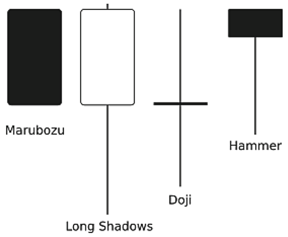


Fig. 2. Some of the most popular Japanese candlesticks.

The remaining paper is organized as follows. Section 2 contains a literature overview. Section 3 is dedicated to a detailed overview of the proposed forecasting model. In Sect. 4 model evaluation and performance metrics are presented. Section 5 presents the conclusions and future work.

2 Related Work

The stock price prediction is very difficult task since many parameters have to be considered, where many of them can not be easily modelled. However, in the last decades researchers proposed various models to help with stock trading decisions.

In literature, the authors use the predictive power of Japanese candlesticks mostly on the basis of expert knowledge and rules that are based on past patterns. Many stock forecasting models have been developed to forecast market price. Lu and Shiu [10] used the four-digit numbers approach to categorize two-day candlestick patterns and tested the approach on Taiwanese stock market. They demonstrated that candlestick analysis has value for investors, what violates efficient markets hypothesis [2]. They found some existing patterns not profitable, and showing two new patterns as profitable.

Kamo and Dagli [6] implemented a study that illustrates the basic candlestick patterns and the standard IF-THEN fuzzy logic model. They employed generalized regression neural networks (GRNN) with rule-based fuzzy gating network. Every GRNN handles one OHLC attribute value, which are then combined to the final prediction with fuzzy logic model. They compared the approach to candlestick method based on GRNN with a simple gating network and it performed better.

Jasemi et al. [5] also used neural networks (NN) for technical analysis of Japanese candlesticks. In their approach NN is not used just to learn the candlestick lines and create a set of static rules, but rather NN continuously analyses input data and updates technical rules. The presented approach yields better results than approach using static selection of rules and input signals. Unfortunately, the authors do not present the data, whether the financial success is obtained in the stock market.

Martiny [11] presented the method that utilizes unsupervised machine-learning for automatically discovering significant candlestick patterns from a time series of price data. OHLC data is first clustered using Hierarchical Clustering, then a Naive Bayesian classifier is used to predict future prices based on daily sequences. The performance of the proposed technique was measured by the percentage of properly triggered sell/buy signals. Although authors in [7] argue that clustering of time-series subsequences is meaningless.

Savić [21] explored the idea of combining Japanese Candlestick language with Natural Language Processing algorithm to implement a basic stock value trend forecasting algorithm. The idea was tested on a sample stock data, where the method achieved promising results. Our work is inspired by the results achieved by Savić.

In this work we present a novel method for forecasting future stock value trends that combines technical analysis method of Japanese candlesticks with deep learning. The proposed model integrates Word2Vec, which is commonly used for the processing of unstructured texts into technical analysis. Word2Vec can find the deep semantic relationships between words in the document. In their work, Zhang et al. [24] confirmed that Word2Vec is suitable for Chinese texts clustering and they also state that Word2Vec shows superior performance in texts classification and clustering in English [12–14]. We have employed the Word2Vec approach in the stock value trend prediction and to the best of our knowledge, none of the existing researches uses Word2Vec for forecasting future stock value trends.

3 Proposed Forecasting Model

The proposed forecasting model that combines a set of machine learning methods in a novel and innovative way. The basic assumption behind the proposed approach is that Japanese candlesticks are not only powerful tool for visualizing OHLC data, but also contain predictive power [5,6,8,10].

Our approach exploits Japanese candlesticks where various sequences are used to forecast the value of a stock. Japanese candlesticks are interpreted as a foundation for stocks' language, i.e. words. A language in general consists of words and patterns of words that can be further grouped into sentences that express some deeper meaning. The proposed model relies on the similarities with the natural language.

The forecasting process starts with a transformation of OHLC data into a simplified language of Japanese candlesticks, i.e. stocks' language. The acquired language is then processed with the NLP algorithm Word2Vec [13] where we train the model with given characteristics and the legality of the proposed stocks' language. The trained model is then used to predict future trends in stock value. The approach is depicted in Fig. 3, with detailed description provided in the following subsections.

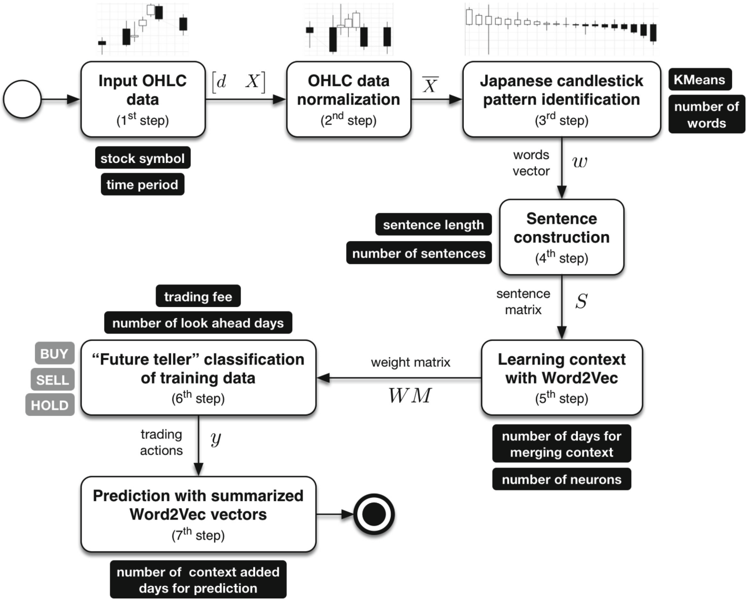


Fig. 3. Steps of proposed forecasting model.

3.1 Input OHLC Data

For a given stock we observe the ***input data*** on a trading day basis for $\boldsymbol{n_d}$ ***trading days*** as defined in the following matrix

$$\left[d_{(1\times n_d)}\ X_{(4\times n_d)}\right] = \left[\begin{array}{c|cccc} d_1 & O_1 & H_1 & L_1 & C_1 \\ d_2 & O_2 & H_2 & L_2 & C_2 \\ \dots & \dots & \dots & \dots & \dots \\ d_{n_d} & O_{n_d} & H_{n_d} & L_{n_d} & C_{n_d} \end{array}\right] \tag{1}$$

where $d_{(1\times n_d)}$ is a vector of trading days and $X_{(4\times n_d)}$ is a matrix of OHLC trading data.

Japanese candlesticks are presented as OHLC tuples, where individual four attributes denote absolute value in time. Raw OHLC data in Eq. 1 are convenient for graphical presentation (see Fig. 4) but are not most suitable for further processing.

3.2 OHLC Data Normalization

Since we are interested in the shape of Japanese candlesticks and not absolute value, the OHLC tuples were normalized by dividing OHLC data attributes (Open, High, Low, Close) with Open attribute as follows

$$norm\big(\langle O, H, L, C\rangle\big) = \left\langle 1, \frac{H}{O}, \frac{L}{O}, \frac{C}{O} \right\rangle : X \rightarrow \overline{X} \tag{2}$$

The employment of transformation from Eq. 2 results in a new input trading data matrix

$$\overline{X}_{(4\times n_d)} = \begin{bmatrix} 1 & \frac{H_1}{O_1} & \frac{L_1}{O_1} & \frac{C_1}{O_1} \\ 1 & \frac{H_2}{O_2} & \frac{L_2}{O_2} & \frac{C_2}{O_2} \\ \dots & \dots & \dots & \dots \\ 1 & \frac{H_{n_d}}{O_{n_d}} & \frac{L_{n_d}}{O_{n_d}} & \frac{C_{n_d}}{O_{n_d}} \end{bmatrix} \tag{3}$$

where the shape of Japanese candlesticks is retained as depicted in Fig. 5, while compared to Fig. 4.

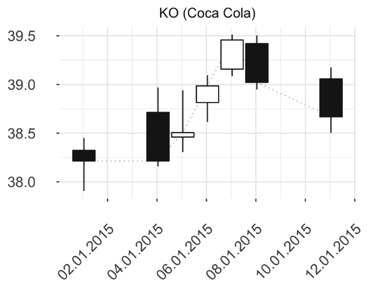


Fig. 4. Raw OHLC data for stock KO in the begining of 2015.

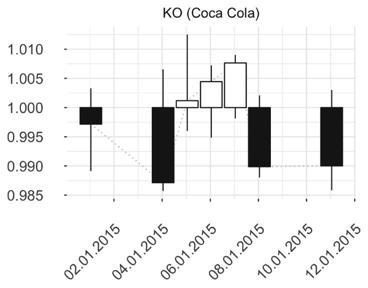


Fig. 5. Normalized OHLC data for stock KO in the begining of 2015.

Figure 4 depicts raw OHLC data, where candlesticks are vertically positioned on the graph corresponding to their relative value. Figure 5 represents the same candlesticks after normalization process that emphasizes and retains the shape of individual candlestick.

3.3 Japanese Candlestick Pattern Identification

Many forecasting models using Japanese candlesticks have a shortcoming of using predefined shapes of candlesticks [11]. Therefore we have adopted the approach of automatically detecting candlestick clusters by employing unsupervised machine learning methods that performed well in previous research [5, 11].

The rationale for using KMeans clustering was to limit the number of possible OHLC shapes (i.e. words of stocks' language) while still being able to influence the unsupervised training process by setting the appropriate threshold for maximum number of different words.

In the process we define the ***maximum number of words*** in stocks' language as $\boldsymbol{n_w}$ and employ ***KMeans clustering*** algorithm to transform input data $\overline{X}$ to vector w as follows

$$KMeans\big(n_w\big) : \overline{X} \rightarrow w \tag{4}$$

where a word w_i is defined by an individual trading day $\overline{X_i}$ and is a representation of a specific Japanese candlestick (the mean value of cluster i). The result of KMeans clustering is a vector

$$w_{(1 \times n_d)} = \big[w_1 \; w_2 \; \ldots \; w_{n_d}\big]^T \tag{5}$$

where given word w_i is an element from a set of all possible Japanese candlesticks, where $i = \big[1, n_w\big]$.

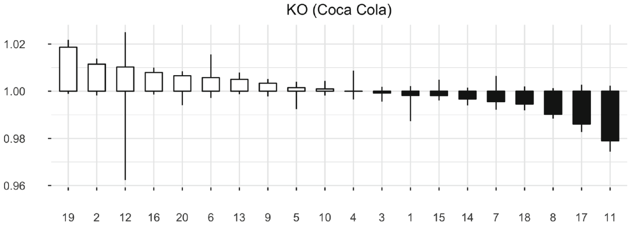


Fig. 6. Example of 20 OHLC pattern clusters for stock KO.

An example of a clustering process for a stock KO (Coca-Cola) is depicted in Fig. 6, where $n_w = 20$ was used for maximum number of words. The value of parameter n_w is based on the Silhouette measure [20], which shows how well an object lies in within a certain cluster (cohesion) compared to other clusters (separation). The Silhouette ranges from -1 to $+1$, where higher value of average Silhouettes means higher clustering validity. In defining stocks' language our aim was also to retain the similarity of words that also exists in natural language by controlling n_w and the Silhouette measure.

3.4 Sentence Construction

With numerous OHLC tuples the potential set of words for the stocks' language is virtually infinite. In the previous section we have limited this to n_w, which directly influences the performance of the proposed predictive model.

Looking at the analysis of past movements in the value of stock we can see that Japanese candlesticks' sequences contain a certain predictive power [10,17]. Therefore, we considered past sequences of OHLC as a basis for the stock trend prediction by forming possible sentences in the future.

The proposed model does not contain any predefined rules for forecasting purposes. Rules used in further processing are created from sequences of patterns that are acquired from past movements in stock value.

We specify a **sentence length $\boldsymbol{l_s}$** that defines the number of consecutive words (i.e. trading days) grouped into sentences. The ***number of sentences $\boldsymbol{n_s}$*** is therefore dependent on the number of trading days n_d and the sentence length l_s and is defined as follows

$$n_s = n_d - (l_s - 1) \tag{6}$$

The result of the sentence construction process is a ***sentence matrix $\boldsymbol{S}$*** of rolling windows of trading data (more specifically words in stocks' language from vector w) from a transformation $w \rightarrow S$. Sentence matrix S with l_s columns (sentence length) and n_s rows (number of sentences) is further defined as

$$S_{(l_s \times n_s)} = \begin{bmatrix} w_1^{'} & w_2^{'} & \dots & w_{l_s}^{'} \\ w_2^{'} & w_3^{'} & \dots & w_{l_s+1}^{'} \\ \dots & \dots & \dots & \dots \\ w_{n_s}^{'} & w_{n_s+1}^{'} & \dots & w_{n_d}^{'} \end{bmatrix} \tag{7}$$

At first glance, such OHLC language seems very simple. However, considering the number of possible values for each word w_i, a set of different possible sentences or patterns is enormous. We believe that the language thus defined has a high expressive power and is suitable for predictive purposes.

3.5 Learning Context with Word2Vec

Based on the patterns in OHLC sentences, the model builds the language context that is then used to perform predictions in the following steps. The system employs historical data, recognizes existing patterns in sentences, learns the context of the words and also renews the context according to new acquired data by employing ***Word2Vec algorithm*** [13] for training the context.

Word2Vec algorithm with skip-gram [12,13] uses a model to represent words with vectors from large amounts of unstructured text data. In the training process, Word2Vec acquires vectors for words that explicitly contain various linguistic rules and patterns by employment of neural network that contains only one hidden level, so it is relatively simple. Many of these patterns can be represented as linear translations. The Word2Vec algorithm has proved to be an excellent tool for analysing the natural language, for example, the calculation

$$\mathit{vector}(\text{`Madrid'}) - \mathit{vector}(\text{`Spain'}) + \mathit{vector}(\text{`Paris'})$$

yields the result that is closer to the vector('France') than any other word vector [12,14].

For learning context in financial trading with Word2Vec we define ***the number of days for merging context $\boldsymbol{n_{ww}}$*** and ***the number of neurons $\boldsymbol{n_v}$*** in hidden layer weight matrix. Word2Vec algorithm performs the following transformation

$$W2V\big(S, n_{ww}, n_v\big) : S \rightarrow WM \tag{8}$$

where the result of Word2Vec learning phase is ***a Weight Matrix $\boldsymbol{WM}$*** with n_v columns (number of vectors) and n_w rows (number of words in stocks' language) and is defined as follows

$$WM_{(n_v \times n_w)} = \begin{bmatrix} v_{1,1} & v_{1,2} & \cdots & v_{1,n_v} \\ v_{2,1} & v_{2,2} & \cdots & v_{2,n_v} \\ \cdots & \cdots & \cdots & \cdots \\ v_{n_w,1} & v_{n_w,2} & \cdots & v_{n_w,n_v} \end{bmatrix} \tag{9}$$

with $v_{i,j}$ as the j-th vector (weight) of word w_i.

3.6 "Future Teller" Classification of Training Data

The proposed model is already capable of using the context that it learned from historical data for creating OHLC predictions. However, our aim is that the predictive model would, based on input OHLC sequence, trigger one of the following actions: (i) BUY, (ii) SELL, (iii) HOLD or do nothing.

For prediction of future stock price we label trading days from matrix X in training set with ***trading actions $\boldsymbol{y}$*** where

$$y_{(1 \times n_d)} = \big[A_1 \; A_2 \ldots A_{n_d}\big]^T \tag{10}$$

and we classify the individual trading day y_i as BUY, SELL or HOLD based on the number of ***look ahead days $\boldsymbol{n_{la}}$*** and ***the trading fee $\boldsymbol{v_{fee}}$*** as follows

$$y_i = \begin{cases} 0 : \text{BUY} & n_{max} \cdot C_j > n_{max} \cdot C_i + 2 \cdot v_{fee}, j \in \big[i, i + n_{la}\big] \\ 1 : \text{SELL} & n_{max} \cdot C_j < n_{max} \cdot C_i - 2 \cdot v_{fee}, j \in \big[i, i + n_{la}\big] \\ 2 : \text{HOLD} & \text{otherwise} \end{cases} \tag{11}$$

where C_i is the stock's close price of a given trading day i and $\boldsymbol{n_{max}} = \left\lceil \frac{e}{C} \right\rceil$ is ***the maximum number of stocks to trade*** with $\boldsymbol{e}$ as ***the initial equity***.

3.7 Prediction

Our proposed model includes classification using the ***SoftMax algorithm*** in our Word2Vec neural network (NN). SoftMax regression is a multinomial logistic regression and it is a generalization of logistic regression. It is used to model categorical dependent variables (e.g. 0 : BUY, 1 : SELL and 2 : HOLD) and the categories must not have any order (or rank).

The output neurons of Word2Vec NN use Softmax, i.e. output layer is a Softmax regression classifier. Based on input sequence, SoftMax neurons will output probability distribution (floating point values between 0 and 1), and the sum of all these output values will add up to 1.

Over-fitting of data may excessively increase the model parameters and may also affect the model performance. In order to avoid over-fitting of our model, we employed least squares regularization that uses cost function which pushes the coefficients of model parameters to zero and hence reduce cost function.

For learning any model we have to omit training days without class prediction, due to look ahead of "Future Teller" from Sect. 3.6, where ***the corrected number of trading days*** is $\overline{\boldsymbol{n_d}} = n_d - n_{la}$.

Basic Prediction. In building a basic prediction we use normalized OHLC data from matrix $\overline{X}$ (see Sect. 3.2) and vector of trading actions y from "Future Teller" classification (see Sect. 3.6), where SoftMax classifier defines the following transformation

$$\left[\overline{X}_{(3\times\overline{n_d})}\ y_{(1\times\overline{n_d})}\right] = \left[\begin{array}{ccc|c} \frac{H_1}{O_1} & \frac{L_1}{O_1} & \frac{C_1}{O_1} & A_1 \\ \frac{H_2}{O_2} & \frac{L_2}{O_2} & \frac{C_2}{O_2} & A_2 \\ \dots & \dots & \dots & \dots \\ \frac{H_{\overline{n_d}}}{O_{\overline{n_d}}} & \frac{L_{\overline{n_d}}}{O_{\overline{n_d}}} & \frac{C_{\overline{n_d}}}{O_{\overline{n_d}}} & A_{\overline{n_d}} \end{array}\right] \rightarrow y = f\left(\frac{H}{O}, \frac{L}{O}, \frac{C}{O}\right) \tag{12}$$

As expected, basic prediction does not perform well as it does not include the context in which OHLC candlesticks appear and influence price movement. Therefore, the following section presents prediction with Word2Vec and taking into account of context by adding previous days OHLC candlesticks.

Prediction with Summarized Word2Vec Vectors. From vector of words w (see Eq. 5) and vector of trading actions y (see Eq. 10) in the following format

$$\left[w_{(1\times\overline{n_d})}\ y_{(1\times\overline{n_d})}\right] = \left[\begin{array}{c|c} w_1 & A_1 \\ w_2 & A_2 \\ \dots & \dots \\ w_{\overline{n_d}} & A_{\overline{n_d}} \end{array}\right] \tag{13}$$

we replace words w_i with a Word2Vec representation with n_v features vector (hyper parameter) from Weight Matrix $WM_{(n_v\times n_w)}$ (see Eq. 9), where $w_i = \left[v_{i,1}, v_{i,2}, \dots, v_{i,n_v}\right]$. Training data in a matrix $X^{'}_{(n_v\times\overline{n_d})}$ is defined as follows

$$\left[X^{'}_{(n_v\times\overline{n_d})}\ y_{(1\times\overline{n_d})}\right] = \left[\begin{array}{cccc|c} v_{1,1} & v_{1,2} & \dots & v_{1,n_v} & A_1 \\ v_{2,1} & v_{2,2} & \dots & v_{2,n_v} & A_2 \\ \dots & \dots & \dots & \dots & \dots \\ v_{w_{\overline{n_d}},1} & v_{w_{\overline{n_d}},2} & \dots & v_{w_{\overline{n_d}},n_v} & A_{\overline{n_d}} \end{array}\right] \tag{14}$$

We add context by ***adding previous $\boldsymbol{n_m}$ trading days*** to the current trading day and define a new input matrix $X^{''}_{(n_v\times\overline{n_d}^{'})}$, where $\overline{n_d}^{'} = \overline{n_d} - n_m$.

Let $\boldsymbol{cv_j} = [cv_{1,j}, cv_{2,j}, \dots, cv_{n_v,j}] \in X^{''}$ be ***a context vector*** for a given trading day j (row j in matrix $X^{''}$), where $j \in [1, \overline{n_d}^{'}]$ and ***contextualized***

input matrix $\boldsymbol{X}^{''}$ is defined as follows

$$\left[X^{''}_{(n_v \times \overline{n_d}')}\ y_{(1\times\overline{n_d}')}\right] = \left[\begin{array}{cccc|c} cv_{1,1} & cv_{1,2} & \dots & cv_{1,n_v} & A_1 \\ cv_{2,1} & cv_{2,2} & \dots & cv_{2,n_v} & A_2 \\ \dots & \dots & \dots & \dots & \dots \\ cv_{w'_{\overline{n_d}},1} & cv_{w'_{\overline{n_d}},2} & \dots & cv_{w'_{\overline{n_d}},n_v} & A_{\overline{n_d}'} \end{array}\right] \tag{15}$$

where context vector cv_j is a sum of vectors of n_m previous trading days as follows

$$cv_j = \sum_{k=j}^{j+n_m} v_k \tag{16}$$

where $v_k = [v_{1,k}, v_{2,k}, \dots, v_{\overline{n_d},k}]$ is the k-th row in matrix $X^{'}$.

4 Evaluation

To measure the quality of our proposed model we have considered various performance metrics and comparative results, based on which we want to evaluate our approach.

Commonly used performance metric is the *Total Hit Ratio* [5,10,15], but it is less adequate to assess model performance in actual trading since it neglects the trading commissions. Another metric that can be used for evaluating performance of the models that predict the actual value of a stock in the future, is *Mean Squared Error (MSE)* [6]. However, our model does not predict the actual value of the stock in the future but merely a general trend (positive, negative, or stagnation), so MSE can not be used. Metrics that are used for classification problems are *classification accuracy*, *AUC*, *logarithmic loss*, etc. [19]. Our model solves multinomial classification problem so the AUC measure [3] is not applicable since it is generally intended for the binary classification. *Classification accuracy* alone can be misleading, so additional measures like precision are required to evaluate a classifier. Logarithmic loss takes into account the uncertainty of prediction based on how much it varies from the actual label. It strongly penalizes the wrong classifications and rewards conservative predictions [3].

We have decided to evaluate our approach using a trading strategy with initial equity and selected prediction model, including trading fees that penalize numerous trading actions which decrease the profitability of prediction model utilization.

The initial equity for every traded stock was \$10.000, with s trading fee of \$15, while the input data was separated into training, test and validation set as depicted in Fig. 7. The historical data included 4.000 OHLC trading days, starting from 1. 5. 2000.

The proposed model was initially evaluated on the shares of Apple (AAPL), Microsoft (MSFT) and Coca-Cola (KO). It yielded promising results, where it outperformed all comparative models on the test set and validation set. However, drawing conclusions based only on three sample shares may not be meaningful,

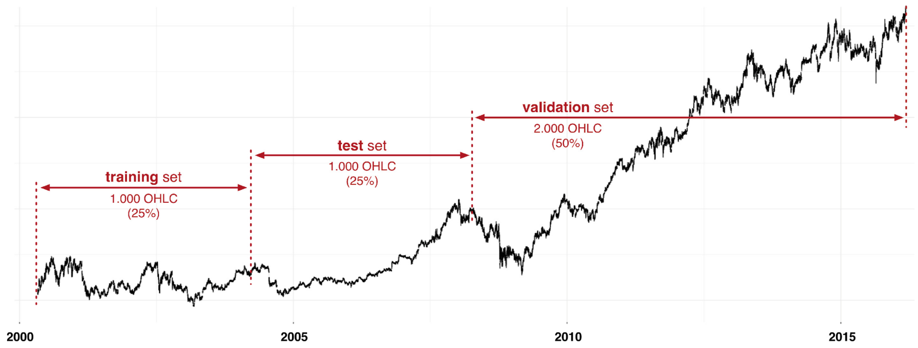


Fig. 7. Separating data into training, test and validation set.

so we carried out extensive testing on a larger data set and performed a confirmatory data analysis.

For the final test set we selected stocks from ***Russell Top 50 Index***, which includes 50 stocks of the largest companies (based on a combination of market cap and current index membership) in the U.S. stock market. The forecasting model was tested for each stock separately. Thus, for each of the 50 stocks, the prediction model was trained based on past stock values of the particular stock. In the test phase, the model parameters were adjusted that the model achieved highest yield for a particular stock. The trained model with parameters tuned for the particular stock was then evaluated on validation set.

Table 1 shows average yield achieved by the proposed ***W2V model*** as well as yield achieved by comparative models (***Buy & Hold***, ***Moving Average (MA)*** and ***MACD***) for the test and validation phase. In the test phase, average yield of the proposed W2V model was much higher than yield of the comparative models.

However, in the validation phase the results were not as good as in the test phase. The average yield of MA and MACD models were still smaller but much closer to the yield of the proposed model, while Buy and Hold outperformed our model. It still has to be noted that the proposed model achieved a positive result in all scenarios.

Table 1. Average yields of forecasting models for the stocks of the Russell Top 50 index at an initial equity of $10.000.

	Buy & Hold	MA (50,100)	MACD	W2V
Test phase	$2,818.98	$1,073.06	–$482.04	$11,725.25
Validation phase	$16,590.83	$6,238.43	$395,10	$10,324.24

In the test phase our model generates profit for all except one stock (i.e. JNJ), where zero profit is achieved. What is more, our model outperformed the comparative models in all but two cases (stocks SLB, JNJ). In the validation phase the results are worse but still encouraging. Only in 14% of cases the model outputs negative yield, while in 16% of cases the model outperformed all comparative models. In 30% of cases the model was the second best model. What is more, in 7 cases the model's yield was very close to the yield of the best method.

In order to obtain statistically significant results we carried out Wilcoxon signed-rank test. The null hypothesis for this test is that the medians of two samples are equal (e.g. Buy & Hold vs. W2V). We accept our hypothesis for p-values which are less than 0.05.

Table 2. The Wilcoxon Signed Rank Test for forecast models.

	Buy & Hold		MA (50,100)		MACD	
	W	*p-value*	*W*	*p-value*	*W*	*p-value*
Test phase	2	<.0001	1	<.0001	1	<.0001
Validation phase	-	-	427	<.0210	155	<.0001

Table 2 depicts values for test statistics *W* and *p-values*. If we focus on the test phase, obtained *p-values* are much lower than 0.05 for all three popular models. This means that there is a statistically significant difference between the resulting yields achieved by our proposed model and existing three models. For the test phase we can conclude with a high level of confidence that appropriately parametrised proposed model W2V performed better than existing three models. As mentioned earlier, the proposed model achieved slightly worse results in the validation phase.

In the validation phase the difference in returns between the proposed model and reference models was statistically significant only for MACD and MA. When compared to Buy & Hold, the W2V method yields lower returns. That can be seen already from the average yields in the Table 1.

The results of the proposed approach demonstrated that with the correct selection of parameters our model achieves statistically significantly better yields than the reference popular methods.

5 Conclusion and Future Work

Our research focused on forecasting trends in stock values. In this paper, we developed a novel approach for stock trend prediction and tested it for financial success rather than just focusing on prediction accuracy. To conduct the experiments, we selected three sample stocks – Apple (AAPL), Coca-Cola (KO) and Microsoft (MSFT) – while confirmation analysis was performed with analysis on Russell Top 50 Index.

We realized that even if the forecasting model has high prediction accuracy, it can still achieve bogus financial results, if poor trading strategy is used. A detailed analysis of the proposed forecast models in the testing phase revealed that despite the simplicity its performance was very good with statistical significance.

A more detailed analysis of trading graphs and statistical analysis showed that the proposed model has a great potential for practical use. However, it is too early to conclude that the proposed model provides a financial gain, as we have shown that selected model parameters are not equally appropriate for different time periods in terms of yield. We have also shown that the forecast model is strongly influenced by the training data set. If the model is trained with data that contains bear trend, the predictive model might be very cautious despite the general growth trend of validation data set. The problem is due to over-fitting, so training with more data would help. Some of the state-of-art machine learning algorithms like *Word2Vec* are dependent on a large-scale data set to become more efficient and eliminate the risk of over-fitting.

There is still room for improvement in the trading strategy. In the future, we would like to incorporate the stop loss function and already known and proven technical indicators. Future improvements also include the use of OHLC data of other stocks in the training phase as we acquire more diverse patterns that helps algorithms to detect the underlying pattern better. To improve classification accuracy and logarithmic loss, the SoftMax algorithm could also be replaced with advanced machine learning classification algorithms. One of the alternative methods of forecasting, which would be worth exploring in the future, would be a simple linear operation of aggregating vector representations of the last n Japanese candlesticks. This way we could obtain a daily, weekly or monthly trend forecast.

References

1. Abad, C., Thore, S.A., Laffarga, J.: Fundamental analysis of stocks by two-stage dea. Manag. Decis. Econ. **25**(5), 231–241 (2004)
2. Fama, E.F.: Efficient Markets Hypothesis. Ph.D. thesis, Ph.D. dissertation, University of Chicago Graduate School of Business (1960)
3. Fawcett, T.: An introduction to roc analysis. Pattern Recogn. Lett. **27**(8), 861–874 (2006)
4. Huang, Y., et al.: Exploiting twitter moods to boost financial trend prediction based on deep network models. In: Huang, D.-S., Han, K., Hussain, A. (eds.) ICIC 2016. LNCS (LNAI), vol. 9773, pp. 449–460. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-42297-8_42
5. Jasemi, M., Kimiagari, A.M., Memariani, A.: A modern neural network model to do stock market timing on the basis of the ancient investment technique of japanese candlestick. Expert Syst. Appl. **38**(4), 3884–3890 (2011)
6. Kamo, T., Dagli, C.: Hybrid approach to the japanese candlestick method for financial forecasting. Expert Syst. Appl. **36**(3), 5023–5030 (2009)
7. Keogh, E., Lin, J.: Clustering of time-series subsequences is meaningless: implications for previous and future research. Knowl. Inf. Syst. **8**(2), 154–177 (2005)

8. Lu, T.H.: The profitability of candlestick charting in the taiwan stock market. Pac.-Basin Finan. J. **26**, 65–78 (2014)
9. Lu, T.H., Shiu, Y.M.: Pinpoint and synergistic trading strategies of candlesticks. Int. J. Econ. Finan. **3**(1), 234 (2011)
10. Lu, T.H., Shiu, Y.M.: Tests for two-day candlestick patterns in the emerging equity market of taiwan. Emerg. Markets Finan. Trade **48**(sup1), 41–57 (2012)
11. Martiny, K.: Unsupervised discovery of significant candlestick patterns for forecasting security price movements. In: KDIR, pp. 145–150 (2012)
12. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
13. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems, pp. 3111–3119 (2013)
14. Mikolov, T., Yih, W.T., Zweig, G.: Linguistic regularities in continuous space word representations. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 746–751 (2013)
15. Ming, F., Wong, F., Liu, Z., Chiang, M.: Stock market prediction from WSJ: text mining via sparse matrix factorization. In: 2014 IEEE International Conference on Data Mining (ICDM), pp. 430–439. IEEE (2014)
16. Nassirtoussi, A.K., Aghabozorgi, S., Wah, T.Y., Ngo, D.C.L.: Text mining of news-headlines for forex market prediction: a multi-layer dimension reduction algorithm with semantics and sentiment. Expert Syst. Appl. **42**(1), 306–324 (2015)
17. Nison, S.: Japanese Candlestick Charting Techniques: A Contemporary Guide to the Ancient Investment Techniques of the Far East. New York Institute of Finance, New York (1991)
18. do Prado, H.A., Ferneda, E., Morais, L.C., Luiz, A.J., Matsura, E.: On the effectiveness of candlestick chart analysis for the brazilian stock market. Procedia Comput. Sci. **22**, 1136–1145 (2013)
19. Read, J., Pfahringer, B., Holmes, G., Frank, E.: Classifier chains for multi-label classification. Mach. Learn. **85**(3), 333 (2011)
20. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. J. Comput. Appl. Math. **20**, 53–65 (1987)
21. Savić, B.: Tvorba jezika japonskih svečnikov in uporaba NLP algoritma Word2Vec za napovedovanje trendov gibanja vrednosti delnic. Master's thesis, University of Ljubljana, Faculty of Computer and Information Science, Ljubljana, Slovenia, July 2016. https://repozitorij.uni-lj.si/IzpisGradiva.php?id=87581&lang=slv
22. Shynkevich, Y., McGinnity, T.M., Coleman, S.A., Belatreche, A.: Forecasting movements of health-care stock prices based on different categories of news articles using multiple kernel learning. Decis. Support Syst. **85**, 74–83 (2016)
23. Taylor, M.P., Allen, H.: The use of technical analysis in the foreign exchange market. J. Int. Money Finan. **11**(3), 304–314 (1992)
24. Zhang, D., Xu, H., Su, Z., Xu, Y.: Chinese comments sentiment classification based on word2vec and SVMperf. Expert Syst. Appl. **42**(4), 1857–1863 (2015)

The Algorithm for Constrained Shortest Path Problem Based on Incremental Lagrangian Dual Solution

Boris Novikov and Roman Guralnik(✉)

Saint Petersburg State University, Universitetskaya emb. 7/9,
199034 Saint-Petersburg, Russia
romasha_nar@mail.ru

Abstract. Most of the systems that rely on the solution of shortest path problem or constrained shortest demand real-time response to unexpected real world events that affect the input graph of the problem such as car accidents, road repair works or simply dense traffic. We developed new incremental algorithm that uses data already present in the system in order to quickly update a solution under new conditions. We conducted experiments on real data sets represented by road graphs of the cities of Oldenburg and San Joaquin. We test the algorithm against that of Muhandiramge and Boland [1] and show that it provides up to 50% decrease in computation time compared to solving the problem from scratch.

Keywords: Incremental · Constrained shortest path · Road graphs

1 Introduction

With graph databases being one of the central representations of big data, we focus our attention on their dynamic version in the form of dynamic road graphs. Calculating a pathway between two points is an essential combinatorial optimization problem not only as a stand-alone problem, but also as a subtask for a series of other more complex optimization problems. The list of examples includes the travelling salesman problem and its counterpart – vehicle routing problem [2, 3]. In the latter problem a route is constructed from several points of interest. In case these points are represented by some graph nodes, it is almost always assumed that the distance between two points is designated by the graph shortest path between them. Other problems include detecting arbitrage opportunities in currency markets, pathfinding problems that are used, for instance, by AI (artificial intelligence) to plot routes or by video game engines to assist users in plotting route.

Several complex applications of shortest path problem require to approach real life conditions and to consider the resource constrained shortest path problem (RCSPP), where graph edges, besides edge cost, are also associated with the resource, which is consumed upon travelling through this edge. The solution path summary resource consumption should fall within the range of $[0, W_L]$. A wide set of problems which require solving constrained shortest path as a subtask include problems of long-haul

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 360–375, 2018.
https://doi.org/10.1007/978-3-319-97571-9_28

aircraft and truck routing [4], military path planning under resource constraints [5], crew scheduling problems [6], pipeline and valve location [7], designing telecommunication network with relay nodes [8]. RSCPP was extended by Smith et al. [9], introducing replenishment edges, which reset consumed resource to zero upon travelling a replenishment edge. Such setting is convenient and used in aircraft routing.

To get even closer to real life conditions we decided to consider constrained shortest path problem applied to real traffic events that may occur on the road. These events may include regular traffic jams, car accidents, road line repair works, road closures etc. We assume that we have calculated (by using an algorithm that we will refer to as *the baseline* algorithm, which implies that we will propose another one that in some way improves this baseline algorithm) a constrained shortest path from desired a source to a desired destination and then one (or several) of the mentioned events take place. All of these events imply that car flow through this road is impaired, that is the number of cars that pass through this segment of road per particular unit of time as well as average speed of travelling through the segment are reduced.

To model this behavior of the system we use resource constrained shortest path setting, where the cost of the edge denotes L_2 distance of the road segment and the weight of the edge represents traffic flow with inverse proportion (the greater the weight of the edge – the less is the flow through this edge). This means that we have to recalculate our solution path from the source to the destination but now taking into account that some of the edges have increased their weight. In cases where the changes in graph edge weights are not heavy (i.e. only a small amount of edges have changed their weights) but still affect our optimal solution it may prove highly inefficient to perform all calculations from scratch. With that in mind, we developed new incremental algorithm to utilize data obtained during the initial run of the baseline algorithm.

Our algorithm is influenced by the algorithm of Muhandiramge and Boland [1] and utilizes the similar preprocessing method, when the solving of Lagrangian dual is integrated with network reduction. Moreover, we combine the above-mentioned algorithm with modified version of the algorithm of Pallottino and Scutella [10] for reoptimizing shortest path trees. This will allow us to use already calculated trees to find new trees with changed edge weights.

2 Related Work

Resource constrained shortest path problem has been extensively studied for several past decades. In most of the works the optimal solution for RCSPP is obtained in three step approach: preprocessing, network reduction and gap closing. The work by Aneja et al. [11] was published in 1983 and is considered to be the first to utilize preprocessing. Its main idea lies in removing nodes and edges that cannot be a part of optimal solution by checking for feasibility every path through considered node or edge.

Several more recent papers use the above-mentioned three step approach. Beasly and Cristofides [12] apply the similar preprocessing procedure but also perform node checks with reduced cost and best lower bound obtained by solving the Lagrangian dual. Dumitrescu and Boland [13] used the same preprocessing but considered regular costs instead of Lagrangian relaxed costs. They conducted testing on sparse network and achieved heavy reductions in the size of the graph. Mehlhorn and Ziegelmann [14] suggest an algorithm with more effective preprocessing due to obtaining better upper bound.

Muhandiramge and Boland [1] were the first to suggest an algorithm that combines preprocessing and Lagrangian dual solving in one step, thus offering a two-step approach. To solve Lagrangian dual Kelley's Cutting Plane Method (KCPM) is used on the set of all Lagrange multipliers. In the gap closing phase a modified version of Carlyle's and Wood's [15] enumeration is utilized and represents a depth-first branch and bound search that uses shortest path trees for optimal Lagrange multiplier to perform tests and fathom branches. Muhandiramge and Boland [1] also offer an SPT reoptimization algorithm that recalculates SPT if it is necessary. Their reoptimization algorithm employs Dijkstra's shortest path algorithm, starting from some advanced point and does not consider changes in edge weights.

As for incremental approaches in regular and constrained shortest path, several algorithms have been suggested in the past 30 years. Gallo [16] proposed the first algorithm to recalculate shortest paths but only for particular cases when the source vertex has changed or exactly one edge has lowered its cost. Ramalingam and Reps [17], King and Thorup [18], Demetrescu [19] provided several reoptimization algorithms for cases when exactly one edge change its weight (in any direction). Pallotino and Scutella [10] devised a method to reoptimize single-source shortest path tree in two-phase approach, dealing with edges that increase and decrease their weights separately. The work by Zhu [20] provides reoptimization algorithm for shortest path tree in an acyclic graph. They consider a case where RCSPP is a subproblem in column generation, i.e. edges change their costs instead of weights as in our problem definition. Our algorithm also deals with graph reductions that is when we need to update shortest path tree on the graph with different number of vertices.

3 Problem Definition

Let $G = (V, A)$ be a directed graph, where V is the set of nodes and A is the set of edges. We give every node a label $i \in Z^+$ - a set of positive integers, so every edge by default gets labeled as (i, j) with i being the source and j being the target of the edge. We denote with s and t the source and target for the whole problem.

Every edge of the graph is associated with two real non-negative values: cost and weight. This can be described with functions $C : A \rightarrow R^+$ and $W : A \rightarrow R^+$. We note here that our algorithm can be easily applied to the case with $W : A \rightarrow (R^+)^n$.

Path p from node i_1 to node i_k is a sequence of edges $(i_1, i_2), \ldots, (i_{k-1}, i_k)$ s.t. $(i_{l-1}, i_l) \in A, \forall l = 1, \ldots, k$. We refer to the cost and weight of the path p as $C(p)$ and

$W(p)$ respectively. If we denote with P_G the set of all paths from s to t then the regular RCSPP would imply the search for such path p^* that

$$C(p^*) = \min_{p \in P_G} C(p) \tag{1}$$

$$\text{s.t. } W(p) \leq W_L \tag{2}$$

where W_L is the global weight limit for the path, designated by problem input.

Next, we suppose that constrained shortest path problem was solved with weight function W and we have to apply now the new weights to graph edges. Due to the way the definitions are set, we don't have to apply changes to the graph itself in order to introduce incrementality, but can rather say that we still have the same graph G and new weight function $W^{'} : A \rightarrow R^{+}$. Considering real life everyday traffic events, we utilize the following statement:

$$\forall e \in AW^{'}(e) \geq W(e). \tag{3}$$

The main idea of incremental RCSPP is formulated as follows. Suppose we solved the RCSPP problem stated with weight function W with some baseline algorithm. We propose a new incremental algorithm for RCSPP to solve the problem under new weight function W' using data obtained with the baseline algorithm. The proposed algorithm is faster than the one that solves the new problem from scratch.

4 Lagrangian Dual and Baseline Algorithm

Our incremental algorithm utilizes the data obtained by the initial run of the base algorithm of Muhandiramge and Boland [1] to give preprocessing a head start. It initializes the problem from advanced point and then continues with solving the Lagrangian dual. This section will provide a brief overview of the Muhandiramge and Boland algorithm in order to explain what data from baseline algorithm run we use and how we do it. For more detailed exposition the reader is referred to [1]. As we mentioned earlier, the algorithm of [1] consists of only two phases – solving the Lagrangian dual and gap closing – since preprocessing (or network reduction) is integrated in the first phase.

4.1 Simple Node Elimination

As for the first phase of [1], there are several options for network reduction. We decided to dwell on the basic part of it called Simple Node Elimination (SNE) and to create its incremental counterpart.

To set up a Lagrangian dual problem for RCSPP a relaxed weight constraint function is introduced

$$\mathcal{L}(\lambda, p) = C(p) + \lambda(W(p) - W_L), \tag{4}$$

where coefficient $\lambda \geq 0$ is known as Lagrange multiplier. For the sake of brevity the *reduced cost function* is sometimes introduced as

$$\Lambda(\lambda, a) = (C + \lambda W)(a) \tag{5}$$

for every edge $a \in A$ and

$$\Lambda(\lambda, p) = \sum\nolimits_{a \in p} (C + \lambda W)(a) \tag{6}$$

for every path p.

Next $\mathcal{L}(\lambda, p)$ is minimized over $p \in P_G$ to obtain Lagrangian dual function

$$\Phi_G(\lambda) = \min\nolimits_{p \in P_G} \Lambda(\lambda, p) - \lambda W_L. \tag{7}$$

It is known that for every $\lambda \geq 0$ Lagrangian dual function provides a lower bound for the value of the target primal function, hence we want to maximize $\Phi_G(\lambda)$ over all non-negative lambdas to obtain greatest lower bound. For a particular value of λ the value of $\Phi_G(\lambda)$ can be calculated by solving unconstrained shortest path problem. By construction, $\Phi_G(\lambda)$ is a piecewise concave function so the authors of [1] rely on modified Kelley's cutting plane method (KCPM). They introduce the convenience notation of minimum cost path form i to j with respect to reduced cost function Λ, found by shortest path calculator, and denote it as Q_{ij}^{λ}. Q_{ij}^{∞} denotes the minimum weight path. Detailed discussion on KCPM is provided in [1].

To combine preprocessing and solving of the Lagrangian dual Muhandiramge and Boland [1] integrate KCPM with network reduction in the following way.

First, an initialization procedure is performed that sets $\lambda^{+} = 0$ and $\lambda^{-} = \infty$ (values, maintained by KCPM) and for each value calculates its forward and reverse shortest path tree. Forward shortest path tree (SPT) is a tree that stores a shortest path from source vertex (vertex *s*) to every other vertex in the network, reverse SPT stores paths from every vertex to target vertex (vertex *t*). This procedure not only verifies that the problem is feasible and non-trivial but finds a feasible solution (if there are any) represented by minimum weight path. Since every feasible solution provides an upper bound whereas Lagrangian dual value provides a lower bound, network reduction can be performed for every node independently. Suppose we have $P_{G,k}$ – a set of all paths from s to t through vertex k. Now let us consider current problem upper bound U and a Lagrangian dual

$$\Phi_{G,k}(\lambda) = \min\nolimits_{p \in P_{G,k}} \Lambda(\lambda, p) - \lambda W_L. \tag{8}$$

If for some $\lambda \geq 0$ it appears that $\Phi_{G,k}(\lambda) \geq U$ then node k can be eliminated from the graph with all of its adjacent edges. It is important to note that for a certain value of λ its forward and reverse SPTs provide all shortest paths through each vertex of the graph (to get such a path through k we can concatenate the one from s to k. obtained from forward SPT with that from k to t obtained from reverse SPT).

The algorithm inserts the elimination checks for every vertex after calculation of λ_{new} and its SPTs. To increase elimination efficiency, this step is preceded by traversing the graph in search of feasible paths at every node in an attempt to improve current upper bound. Then lower bound is calculated for every node and the elimination checks are performed.

We state here full pseudo-code of SNE as provided by Muhandiramge and Boland. We use steps from 3 to 8 (as well as incremental SPT update) to continue solving KCPM for our incremental preprocessing, described in Sect. 7.

```
1.  Do $initialization\ procedure$ (check if the problem is feasible and
    non-trivial and of so: set $U = C(Q_{st}^{\infty})$ and $pU = Q_{st}^{\infty}$ where $pU$ is the
    upper bound path)
2.  $\lambda^{+} = 0$
    $\lambda^{-} = \infty$
    $L = (\lambda^{+}, \lambda^{-})$
3.  $\lambda_{new} = \frac{C\left(Q_{st}^{\lambda^{+}}\right) - C\left(Q_{st}^{\lambda^{-}}\right)}{W\left(Q_{st}^{\lambda^{-}}\right) - W\left(Q_{st}^{\lambda^{+}}\right)}$
    $\mathcal{L}_{new} = C\left(Q_{st}^{\lambda^{+}}\right) + \lambda_{new}\left(W\left(Q_{st}^{\lambda^{+}}\right) - W_L\right)$

$Calculate\ forward\ and\ reverse\ SPTs, \Phi_G, \Phi'_G\ for\ \lambda_{new}, add\ \lambda_{new}\ to\ L$
4.  For $each\ node\ k \in V,$
        If $C\left(Q_{sk}^{\lambda_{new}} + Q_{kt}^{\lambda_{new}}\right) < U\ and\ W\left(Q_{sk}^{\lambda_{new}} + Q_{kt}^{\lambda_{new}}\right) \leq W_L$
            $U = C\left(Q_{sk}^{\lambda_{new}} + Q_{kt}^{\lambda_{new}}\right)$
            $pU = Q_{sk}^{\lambda_{new}} + Q_{kt}^{\lambda_{new}}$ (new upper bound has been found)
5.  $V' = \emptyset\ \ and\ A' = \emptyset$ (vertices and edges that will be deleted)
    For $each\ node\ k \in V$
        If $\Phi_{G,k}(\lambda) \geq U,$
            $V' = V' \cup \{k\}$
            $A' = A' \cup \{all\ edges\ adjacent\ to\ k\}$
      $V = V \backslash V' and\ A = A \backslash A'$
      $G = (V, A)$
6.  If $V = \emptyset,$ STOP (the current upper bound is the optimal solution to
    RCSPP)
    If $\Phi_G(\lambda_{new}) = \mathcal{L}_{new},$ STOP (OLM found)
7.  If $\Phi'_G\ (\lambda_{new}) \leq 0$
        $\lambda^{-} = \lambda_{new}$
        If $SPTs\ for\ \lambda^{+}\ are\ out\ of\ date, recalculate\ SPTs\ for\ \lambda^{+}$
        If $\Phi'_G\ (\lambda^{+}) \leq 0\ and\ \lambda^{+} \neq 0,$
```

```
            Repeat
                $Set\ \lambda^+ to\ the\ next\ lowest\ value\ of\ \lambda\ in\ L$
                $Recalculate\ its\ SPTs\ if\ out\ of\ date$
            Until $either\ \Phi'_G(\lambda^+) > 0\ or\ \lambda^+ = 0$
        If $\lambda^+ = 0\ and\ \Phi'_G(0) \le 0$, STOP (current upper bound is optimal)
    Else
        $\lambda^+ = \lambda_{new}$
        If $SPTs\ for\ \lambda^-\ are\ out\ of\ date, recalculate\ SPTs\ for\ \lambda^-$
        If $\Phi'_G(\lambda^-) > 0\ and\ \lambda^- \ne \infty$,
            Repeat
                $Set\ \lambda^- to\ the\ next\ highest\ value\ of\ \lambda\ in\ L$
                $Recalculate\ its\ SPTs\ if\ out\ of\ date$
            Until $either\ \Phi'_G(\lambda^-) \le 0\ or\ \lambda^- = \infty$
        If $\lambda^- = \infty\ and\ \Phi'_G(\lambda^-) > 0$, STOP (no feasible path in current
  network, current UB is optimal)
  8. Goto $step$ 3
```

Step 7 is necessary since all of the calculated SPTs (except λ_{new}) become obsolete after the reduction of the network in step 4, which for the new graph may result in $\Phi'_G(\lambda^-) > 0$ or $\Phi'_G(\lambda^+) \le 0$. To address this issue the authors of [1] make use of the ordered set L of all lambdas for which SPTs were calculated. This allows to keep λ^+ and λ^- correct throughout the algorithm. For additional clarifications on Simple Node Elimination the reader is referred to the original article [1].

4.2 Gap Closing

For the second phase, which is supposed to close the gap between lower and upper bounds, Muhandiramge and Boland chose to utilize the depth-first branch and bound approach facilitated by fathom tests. This method starts from the source vertex and iterates through graph with a depth-first approach, building a path from visited edges. The fathom test is applied to every branch (the branch is represented by a certain edge) under consideration and depending on the outcome of the test the branch is pruned, i.e. is removed from further considerations since it cannot be in the optimal path. If dfbb (depth-first branch and bound) managed to reach the target vertex the algorithm checks the cost of the path, which was used to reach the target, and, if necessary (i.e. the cost of this path is less than current upper bound), updates upper bound and the result path.

The fathom test uses a procedure similar to KCPM's search for optimal Lagrange multiplier. Keeping in mind that $\Phi(\lambda)$ is concave and having the ordered set $\overline{L} = (\lambda_0, \ldots, \lambda_1)$ of lambda values for which we have SPTs the algorithm performs a bisection search to find the best available lambda (for which $\Phi(\lambda)$ is the greatest). The explicit pseudo-code for fathom test as well as the gap closing phase is given in [1].

5 Incremental SPT Update

As a starting point we assume that the baseline algorithm finished its work, reduced the network and found an optimal path. As the problem statement claims, we need to apply now the new weight function to our graph (that is, to increase the weights of some edges). Because of this, the reduction of some nodes and edges may prove invalid. That means we cannot use the current graph and have to perform new network reduction on the initial full graph. For the reasons stated above in this research we focus our attention on the first phase in an attempt to perform fast network reduction and optimal Lagrange multiplier search based on the data computed in the course of the baseline algorithm. Due to the article size limitations we omit detailed discussions on the gap closing phase.

As we may infer, the largest computational overhead of SNE lies in computing and recomputing shortest path trees. A* (or Dijkstra's) is suggested as a default SPT calculator, but, for example, Dijkstra's $O(|A| + |\mathrm{V}| \log|V|)$ for a single SPT calculation can prove inefficient for big graphs, considering the fact that all calculations have to be performed in real time and provide solutions as fast as possible. It is important to notice that the authors of [1] also provide their own algorithm for updating SPT. However, this algorithm is designed to only recalculate SPTs after network reduction. This can be regarded as the special case of changing edge weight, when edge weight is set to be equal to ∞ (i.e. delete the edge). For that reason such algorithm cannot be applied to general case of edge weight increase. To address this issue we decided to use slightly modified version of Pallottino and Scutella [10] algorithm for reoptimizing shortest path tree. Below, we will provide a brief overview of the algorithm.

5.1 Incremental SPT Update Algorithm

Since the algorithm operates only with unconstrained edge costs, it is convenient to denote a cost of an edge (i,j) as c_{ij} keeping in mind that every SPT is tied to a certain value of λ, so $c_{ij} = C(i,j) + \lambda W(i,j)$. Dijkstra algorithm also provides us with the potential π_i of each node (the weight of the shortest path from SPT root r to this node) as well as reduced cost $\overline{c}_{ij} = c_{ij} + \pi_i - \pi_j$.

Algorithm also receives as input an SPT $T_r = (V, A_r)$ as well as new costs $c^{'}_{ij}$. For T_r to be a shortest path tree it has to satisfy a complementary slackness condition (CSC), that is $\overline{c}_{ij} = 0, \forall (i,j) \in A_r$. If the new edge costs $\overline{c}'_{ij}$ satisfy CSC for every $(i,j) \in A_r$ then the problem is trivial and T_r is returned as an updated SPT.

Otherwise, forest $F_r = (N, A_F)$ is obtained from T_r by removing all edges that do not satisfy CSC $(\overline{c}'_{ij} > 0)$. That means that $A_F = \left\{(i,j) \in A_r : \overline{c}_{ij} = 0\right\}$. Root subtree $T(r) = (N(r), A(r))$ contains SPT root r and does not need reoptimization. Other subtrees are reconnected to $T(r)$ through a series of "hanging operations" described below.

Consider a cut $\left(N(r), \overline{N}(r)\right)$ of the vertex set, where $\overline{N}(r) = N \backslash N(r)$. Hanging operation defines two sets:

$$A^{+} = \left\{(i,j) \in A : i \in N(r), j \in \overline{N}(r)\right\}, \tag{9}$$

known as the set of *border edges*, and

$$A^E = \left\{(i,j) \in A : i \in \overline{N}(r), j \in \overline{N}(r)\right\}, \tag{10}$$

known as the set of *external edges*.
For every node $j \in \overline{N}(r)$, the following values are calculated

$$\begin{gathered}\delta_j = \min\left\{\overline{c}'_{ij} : (i,j) \in In(j) \ \cap A^+\right\}\alpha_j = \min\left\{\overline{c}'_{ij} : (i,j) \in In(j) \ \cap A^E\right\} \\ \delta_j^+ = \min\{\delta_i : i \in N^+(j)\}\delta_j^- = \min\{\delta_i : i \in N^-(j)\} \\ \alpha_j^+ = \min\{\alpha_i : i \in N^+(j)\}\alpha_j^- = \min\{\alpha_i : i \in N^-(j)\},\end{gathered}$$

where $In(j)$ denotes a set of all incoming edges of j, $N^+(j)$ and $N^-(j)$ denote sets of nodes which are ancestors and descendants of j respectively. These values are assumed to be ∞ if the corresponding set is empty.

Next, *gap* Δ is defined for the cut $\left(N(r), \overline{N}(r)\right)$ as

$$\Delta = \min\left\{\delta_j : j \in \overline{N}(r)\right\} \tag{11}$$

and node w for which $\delta_w = \Delta$.

Finally, node v is considered *hangable* if it satisfies the following inequalities:

$$\delta_v \leq \min\left\{\delta_v^+, \delta_v^-\right\} \tag{12}$$

and

$$\delta_v \leq \Delta + \min\left\{\alpha_{\mathrm{v}}^+, \alpha_{\mathrm{v}}^-\right\} \tag{13}$$

If the conditions are true, node v and its subtree $T(v) = (N(v), A(v))$ are hanged to the root subtree $T(r)$ through the edge $(u, v) \in A^+$, such that $\overline{c}'_{uv} = \delta_v$. After the hanging we have:

$$N(r) := N(r) \ \cup N(v) \tag{14}$$

$$A(r) := A(r) \ \cup A(v) \ \cup (u, v) \tag{15}$$

$$\pi_i := \pi_i + \delta_v, \forall i \in T(v) \tag{16}$$

Not only a set of hangable nodes is never empty, it can also contain multiple vertices and their subtrees can be hanged in parallel.

After hanging all currently hangable vertices, the values of alphas and deltas are updated and new hanging operations can be performed. The algorithm stops when all vertices are in the root subtree. More detailed discussions on SPT update and hanging operations can be found in [21, 22].

6 Incremental SPT Update Complexity in the Scope of RSCPP

Now if we return to RSCP problem and try to apply SPT update algorithm in the form stated above, we may face the following difficulty. The current graph on which we want to have an updated SPT can have different set of vertices, compared to the graph on which input SPT has been calculated. This can happen if either of the graph has undergone reduction in the course of KCPM. Since it will cause problems in the entire algorithm logic, we have to address this issue by considering two cases. First case represents an event when new graph does not contain all the vertices from the old graph, that is $\exists i \in V_{old} : i \notin V_{new}$. Second case represents the opposite event when $\exists i \in V_{new} : i \notin V_{old}$. There is actually a third option for the case when both graphs were reduced and now contain different sets of vertices. However, this case is just a combination of the first two and does not require to be handled independently.

First case does not require much attention since it is rather trivial. During initial forest F_r and root subtree $T(r)$ construction our algorithm deletes not only nodes that do not satisfy CSC but also nodes that are not contained in new graph vertex set V_{new}. The latter nodes (which are not in V_{new}) and there adjacent edges also don't participate in hanging operations.

As for the second case, let us consider nodes that are in V_{new} but not in V_{old}. Let us call them e-nodes (from extra nodes). Obviously such nodes fall into $\overline{N}(r)$ and require to be hanged. Moreover, if a node i in $N(r)$ has an adjacent edge (k, i) and node k is an e-node (node i has an edge incoming from e-node), then node i is deleted from $T(r)$ and is also required to be hanged. The latter action is required to keep optimality of the SPT, since that adjacent e-node can offer a better shortest path, than the current one for the node in question.

It is obvious, that the great number of e-nodes can seriously impair computational efficiency of SPT updated. Default complexity for SPT update through hanging operations is $O(m + Rn)$, where m and n are the numbers of edges and vertices respectively and R is the number of hanging iterations (remember that each hanging iteration can hang several subtrees in parallel). For our modified version this would mean a complexity of $O(m_{new} + Rn_{new})$. If there is many e-nodes the number R will be great as well and in the worst case can be equal to n, making worst case complexity of $O\left(m_{new} + n_{new}^2\right)$ which is worse than computing SPT from scratch with Dijkstra's algorithm. However, in practice, Dijkstra's algorithm outperforms incremental SPT update only when we are trying to update SPT calculated on heavily reduced graph, and the new graph being from almost full to non-reduced at all. Of course, we can think of at least one example when this could happen. As soon as the edge weights have changed, the first step of preprocessing performs a test for the problem being trivial. This test is usually done by calculating SPTs for $\lambda = \infty$ (testing if minimum weight path is feasible, otherwise – the problem is infeasible) and for $\lambda = 0$ (testing if minimum cost path is feasible, and if so – the optimal solution for the problem is the minimum cost path). The difficulty may arise if during baseline algorithm SPTs for $\lambda = \infty$ and $\lambda = 0$ were recalculated for heavily reduced graph. So, to avoid recalculation of SPTs calculated on heavily reduced graph and to facilitate the initialization of the incremental RCSPP our algorithm stores two additional SPTs for $\lambda = \infty$ and $\lambda = 0$ and in particular the ones that were computed on non-reduced graph and performs updates using these SPTs.

Now we can discuss the case, were no e-nodes are present. Of course, in this case we know that $m_{new} \le m_{old}$, $n_{new} \le n_{old}$ and complexity of incremental SPT update (the number of hanging operations R) will depend on the number of edges that has changed their weight. Worst case complexity is still $O\left(m_{new} + n_{new}^2\right)$, but we have to remember here that we consider real life events such as car accidents, road line repair works or road closing etc. Hence, it can be inferred that edge weight changes are not scattered randomly across the graph but rather congregated around particular edges. For incremental SPT update that means that lots of previously calculated SPT subtrees remain intact which allows not only to hang subtrees with many nodes to the root subtree but also to hang many subtrees at once (in parallel) and in the end outperform Dijkstra's algorithm. Indeed, if, for instance in some SPT a path from i to j contains three edges with changed weight and these edges are adjacent, then everything before and after this "congregation" does not need to be recomputed and will be present as is in the updated SPT.

As experiments show, there are certain thresholds on the amount of edges that change their weights. These thresholds signify when it is more efficient to use incremental SPT update than to recalculate SPT from scratch. Due to the lack of article space we shall not dwell on the discussions about exact values for the above-mentioned thresholds and move to the discussion of the algorithm itself.

7 Incremental Preprocessing Pipeline

In this section we will provide a pseudo-code for the preprocessing step and explain its general steps.

To perform efficient preprocessing using the data obtained by baseline algorithm, we would like to start from some advanced point, that is, we would like to use some good upper and lower bounds and use them in network reduction.

We start with checking a problem for being feasible or trivial. Normally we would have to do that by updating SPTs for $\lambda = \infty$ and $\lambda = 0$ using SPTs calculated on the full graph. We can notice though, that an update for $\lambda = 0$ does not have to happen here, if the global weight limit stays the same. This is true because minimum cost path stays the same even after weight changes. Since we work with weight increases, we can infer that new problem is non-trivial if the old problem is non-trivial. The reverse statement, however, is not true. Looking for an upper bound we can always rely on minimum weight path, provided it is feasible, if we have recalculated SPTs for $\lambda = \infty$. However, to obtain better upper bound and to possibly avoid recalculation of SPTs we suggest the following procedure.

During the gap closing phase of baseline algorithm, the depth-first branch and bound (dfbb) approach is used to traverse the graph in search of feasible solutions. Every feasible solution found by dfbb is stored for the use in incremental preprocessing. After applying new weight function to edges, we can check every solution for feasibility. We assume that the path with minimum cost out of all feasible solutions (the one that was optimal for the old graph) is infeasible, because the problem would be trivial otherwise, if this path would retain its optimality. However, if any of the remaining feasible solutions appears to be feasible under new weight function, then we can use it as an upper bound for preprocessing and as an indication of problem feasibility. This will allow us to avoid recalculation of SPTs for $\lambda = \infty$ during initialization.

To perform incremental preprocessing we need to find new optimal Lagrange multiplier, since it could change after applying new weights to the edges, and then perform reduction. However, since we have all the information obtained in the course of baseline algorithm, we have a list of values for λ and their, albeit invalid now, SPTs. For that reason we don't need to start KCPM from scratch, that is with $\lambda^{+} = 0$ and $\lambda^{-} = \infty$. To give our preprocessing a head start we can recalculate SPTs for a particular value of λ_{inc} from the list of available lambdas, perform network reductions (since recalculated SPTs will provide us with a lower bound) and continue KCPM setting $\lambda^{+} = \lambda_{inc}, \lambda^{-} = \infty$ or $\lambda^{-} = \lambda_{inc}, \lambda^{+} = 0$ depending on the sign of $\Phi^{'}(\lambda_{inc})$.

The question may arise: what value for λ_{inc} may be optimal? One seemingly logical option would be to choose OLM obtained by baseline's KCPM. However, it may turn out that SPTs for this value might have been calculated on the graph already heavily reduced, which renders recalculations of such SPTs highly inefficient due to the large number of e-nodes. For that reason in our algorithm we were not using OLM obtained in baseline's KCPM. In the course of our experiments on SNE, we made an interesting observation related to values of λ and graph reductions. Let us denote as q_r the number of nodes that were removed from the graph during r-th iteration of KCPM. We noticed that during several initial iterations of KCPM, network reductions can be trivial, i.e. no nodes and edges are deleted. Next, let us assume that first non-trivial reduction happened at r_{nt}-th iteration and we removed $q_{r_{nt}}$ nodes. We observed, that in large majority of cases after r_{nt}-th iteration the amount of nodes removed during network reduction is always less than $q_{r_{nt}}$, that is

$$\forall r > r_{nt}, q_r < q_{r_{nt}} \tag{17}$$

It means that the largest number of nodes is deleted during first non-trivial reduction. Now if we denote as λ_{nt} the value of lambda, which was used to perform first non-trivial graph reduction, we can be certain, that for λ_{nt} baseline algorithm calculates SPTs on the non-reduced graph. We store these SPTs, as well as the value of λ_{nt} right after first non-trivial reduction takes place, since we cannot be sure that these SPTs would not be recalculated on some reduced version of the graph in the course of KCPM.

Thus, in our incremental preprocessing we set $\lambda_{inc} = \lambda_{tr}$ and recalculate SPTs using specific stored ones. After that network reduction is performed using the lower bound obtained from $\Phi(\lambda_{inc})$, set $\lambda^{+} = \lambda_{inc}, \lambda^{-} = \infty$ or $\lambda^{-} = \lambda_{inc}, \lambda^{+} = 0$ depending on the sign of $\Phi^{'}(\lambda_{inc})$, recalculate the necessary invalid SPTs (recalculate SPTs for $\lambda = 0$ if $\Phi^{'}(\lambda_{inc}) \leq 0$ or recalculate SPTs for $\lambda = \infty$ otherwise) and continue KCPM, using incremental SPT updates where necessary.

The initial network reduction with λ_{inc} gives incremental preprocessing a head start and allows to avoid recalculating SPTs for the graphs with a great number of e-nodes.

Next, we provide the pseudo-code for our incremental preprocessing.

```
1.  $Set\ U = \infty$
2.  For $every\ path\ p\ in\ the\ list\ of\ feasible\ solutions\ obtained\ in$
     $the\ gap\ closing\ of\ baseline\ algorithm$
            If $W(p) \leq W_L$
                If $C(p) < U$
                      $U = C(p)$
                      $pU = p$
3.  If $U = \infty$ (no path remained feasible and $U$ was not updated)
        $recalculate\ SPTs\ for\ \lambda = \infty$
        $U = C(Q_{st}^{\infty})$
        $pU = Q_{st}^{\infty}$
4.  $Use\ stored\ full\ SPTs\ to\ recalculate\ SPTs\ for\ \lambda_{inc} = \lambda_{nt}$
5.  For $each\ node\ k \in V,$
        If $C\left(Q_{sk}^{\lambda_{inc}} + Q_{kt}^{\lambda_{inc}}\right) < U\ and\ W\left(Q_{sk}^{\lambda_{inc}} + Q_{kt}^{\lambda_{inc}}\right) \leq W_L$
            $U = C\left(Q_{sk}^{\lambda_{inc}} + Q_{kt}^{\lambda_{inc}}\right)$
            $pU = Q_{sk}^{\lambda_{inc}} + Q_{kt}^{\lambda_{inc}}$ (new upper bound has been found)
6.  $V' = \emptyset\ \ and\ A' = \emptyset$ (vertices and edges that will be deleted)
    For $each\ node\ k \in V$
        If $\Phi_{G,k}(\lambda) \geq U,$
            $V' = V' \cup \{k\}$
            $A' = A' \cup \{all\ edges\ adjacent\ to\ k\}$
      $V = V \backslash V' and\ A = A \backslash A'$
      $G = (V, A)$
7.  If $V = \emptyset$, STOP (the current upper bound is the optimal solution to
RCSPP)
8.  If $\Phi_G'\ (\lambda_{inc}) \leq 0$
        $recalculate\ SPTs\ for\ \lambda = 0$
        $continue\ KCPM\ with\ \lambda^{+} = 0\ and\ \lambda^{-} = \lambda_{inc}$
    Else
        $recalculate\ SPTs\ for\ \lambda = \infty$ (if it was not recalculated in STEP 3)
        $continue\ KCPM\ with\ \lambda^{+} = \lambda_{inc}\ and\ \lambda^{-} = \infty$
```

Thus, our preprocessing algorithm starts from advanced points, performs necessary network reductions and stops when either found optimal solution or optimal Lagrange multiplier for the new problem.

8 Experiments

We conducted extensive experiments on real road graphs of the city of Oldenburg containing 6000 nodes and 14000 edges and the city of San Joaquin [23] containing 18000 nodes and 46000 edges. We used L2 distance of the edge as its cost. To remain close to real life cases we decided to apply the edge weights according to real traffic distribution. We set the edge weight according to the load it is likely to have, which is determined by the number of incoming and outgoing edges of the source vertex of the edge as well as the weights of the outgoing edges, and the number of outgoing edges of the target vertex of the edge. The rationale behind this is simple. Let us denote as $In(i)$ and $Out(i)$ the sets of incoming and outgoing edges of the node i. Now let us consider the edge (i,j). If there is a lot of edges incoming in i, then there is a lot of traffic congregated at that vertex i. This traffic is then distributed across outgoing edges. However, each edge from $e \in In(i)$ "receives" only portion of traffic, determined by the number of outgoing edges of the target vertex of e. Therefore, for the result formula of the edge weight we have

$$W(i,j) = |In(i)| \frac{|Out(j)|}{\sum_{e \in Out(i)} |Out(e_t)|} \tag{18}$$

where e_t denotes the target vertex of the edge e and $|\cdot|$ denotes cardinality of the set. Note that $Out(j)$ is never empty, since we consider two-way roads, so we can always make a U-turn. It is possible to implement the edge weight setting in a way to compute all edge weights in $O(m)$.

We tested our algorithm against that of Muhandiramge and Boland after applying different amount of changes to the edge weights. The only restriction for weight changes was keeping the problem non-trivial, that is changed edges were to affect the solution path. The results are shown in Table 1. Each number represents an average computation time of several runs with different source and target vertices.

Our incremental version provides up to 50% decrease in computation time compared to Simple Node Elimination combined with gap closing. It is important to mention that this time decrease can help in keeping the computations in real-time, which is the purpose of this algorithm. During the day in the big after numerous traffic events this saved time can accumulate in even more impactful difference. Time save comes from faster initialization and recalculations of SPTs and depends on the number of edges that changed their weights.

Table 1. Experimental results with various numbers of e-nodes

Number of e-nodes	Algorithm used	Roads of Oldenburg, s	Roads of San Joaquin, s
Small	Muhandiramge and Boland	1.34	2.13
	Incremental RCSP	0.72	1.18
Medium	Muhandiramge and Boland	1.29	2.20
	Incremental RCSP	1.14	1.96
Large	Muhandiramge and Boland	1.36	1.94
	Incremental RCSP	1.88	2.31

9 Conclusion

In this paper we described a new incremental approach to solving the resource constrained shortest path problem by utilizing the data from the baseline algorithm and starting Lagrangian dual solving from advanced point. We introduced a modified version of unconstrained SPT update for the cases when the graph has either more or less vertices as well as different edge weights. We conducted experiments on real road graphs and achieved decreased computation time compared to recalculating a solution from scratch.

As a part of future work we intend to focus on the gap closing phase and its incremental version and publish already existing progress on the topic. We also plan on analyzing the amount of changes in the graph edge weights. Such analysis can yield important threshold which would signify whether it is more effective to use incremental version, recalculate from scratch or maybe even skip preprocessing phase if the number of changed edges is very low.

References

1. Muhandiramge, R., Boland, N.: Simultaneous solution of Lagrangean dual problems interleaved with preprocessing for the weight constrained shortest path problem. Networks **53**(4), 358–381 (2009). https://doi.org/10.1002/net.20292
2. Guralnik, R.: Incremental rerouting algorithm for single-vehicle VRPPD. In: Proceedings of the 18th International Conference on Computer Systems and Technologies, pp. 44–51. ACM (2017). https://doi.org/10.1145/3134302.3134326
3. Jaw, J.J., Odoni, A.R., Psaraftis, H.N., Wilson, N.H.: A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. Transp. Res. Part B: Methodol. **20**(3), 243–257 (1986). https://doi.org/10.1016/0191-2615(86)90020-2
4. Barnhart, C., et al.: Flight string models for aircraft fleeting and routing. Transp. Sci. **32**(3), 208–220 (1998). https://doi.org/10.1287/trsc.32.3.208
5. Carlyle, W.M., Royset, J.O., Wood, R.K.: Routing military aircraft with a constrained shortest-path algorithm. Naval Postgraduate School, Monterey CA, Department of Operations Research (2007)
6. Desrochers, M., Soumis, F.: A column generation approach to the urban transit crew scheduling problem. Transp. Sci. **23**(1), 1–3 (1989). https://doi.org/10.1287/trsc.23.1.1
7. Laporte, G., Pascoal, M.M.: The pipeline and valve location problem. Eur. J. Ind. Eng. **6**(3), 301–321 (2012). https://doi.org/10.1504/EJIE.2012.046669
8. Cabral, E.A., Erkut, E., Laporte, G., Patterson, R.A.: The network design problem with relays. Eur. J. Oper. Res. **80**(2), 834–844 (2007). https://doi.org/10.1016/j.ejor.2006.04.030
9. Smith, O.J., Boland, N., Waterer, H.: Solving shortest path problems with a weight constraint and replenishment arcs. Comput. Oper. Res. **39**(5), 964–984 (2012)
10. Pallottino, S., Scutella, M.G.: A new algorithm for reoptimizing shortest paths when the arc costs change. Oper. Res. Lett. **31**(2), 149–160 (2003). https://doi.org/10.1016/S0167-6377(02)00192-X
11. Aneja, Y.P., Aggarwal, V., Nair, K.P.: Shortest chain subject to side constraints. Networks **13**(2), 295–302 (1983). https://doi.org/10.1002/net.3230130212
12. Beasley, J.E., Christofides, N.: An algorithm for the resource constrained shortest path problem. Networks **19**(4), 379–394 (1989). https://doi.org/10.1002/net.3230190402

13. Dumitrescu, I., Boland, N.: Improved preprocessing, labeling and scaling algorithms for the weight constrained shortest path problem. Networks **42**(3), 135–153 (2003). https://doi.org/10.1002/net.10090
14. Mehlhorn, K., Ziegelmann, M.: Resource constrained shortest paths. In: Paterson, Mike S. (ed.) ESA 2000. LNCS, vol. 1879, pp. 326–337. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45253-2_30
15. Carlyle, W.M., Royset, J.O., Wood, R.K.: Lagrangian relaxation and enumeration for solving constrained shortest path problems. Networks **52**(4), 256–270 (2008). https://doi.org/10.1002/net.20247
16. Gallo, G.: Reoptimization procedures in shortest path problem. Rivista di matematica per le scienze economiche e sociali **3**(1), 3–13 (1980). https://doi.org/10.1007/BF02092136
17. Ramalingam, G., Reps, T.: An incremental algorithm for a generalization of the shortest-path problem. J. Algorithms **21**(2), 267–305 (1996). https://doi.org/10.1006/jagm.1996.0046
18. King, V., Thorup, M.: A space saving trick for directed dynamic transitive closure and shortest path algorithms. In: Wang, J. (ed.) COCOON 2001. LNCS, vol. 2108, pp. 268–277. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44679-6_30
19. Demetrescu, C.: Fully Dynamic Algorithms for Path Problems on Directed Graphs. Ph.D. thesis, Department of Computer and Systems Science, University of Rome “LaSapienza” (2001)
20. Zhu, X.: The Dynamic, Resource-Constrained Shortest Path Problem on an Acyclic Graph with Application in Column Generation and Literature Review on Sequence-Dependent Scheduling. Doctoral dissertation, Texas A&M University (2007)
21. Nguyen, S., Pallottino, S., Scutellà, M.G.: A New Dual Algorithm for Shortest Path Reoptimization. In: Gendreau, M., Marcotte, P. (eds.) Transportation and Network Analysis: Current Trends. Applied Optimization, vol. 63, pp. 221–235. Springer, Boston (2002). https://doi.org/10.1007/978-1-4757-6871-8_14
22. Pallottino, S., Scutellà, M.G.: Dual algorithms for the shortest path tree problem. Networks **29**(2), 125–133 (1997). https://doi.org/10.1002/(SICI)1097-0037(199703)29:2<125::AID-NET7>3.0.CO;2-L
23. Li, F., Cheng, D., Hadjieleftheriou, M., Kollios, G., Teng, S.-H.: On trip planning queries in spatial databases. In: Bauzer Medeiros, C., Egenhofer, Max J., Bertino, E. (eds.) SSTD 2005. LNCS, vol. 3633, pp. 273–290. Springer, Heidelberg (2005). https://doi.org/10.1007/11535331_16

Current Perspectives on the Application of Bayesian Networks in Different Domains

Galina M. Novikova(✉) and Esteban J. Azofeifa

RUDN University, Moscow, Russia
novikova_gm@rudn.university, esteban.azofeifa@gmail.com

Abstract. Bayesian networks are powerful tools for representing relations of dependence among variables of a domain under uncertainty. Over the last decades, applications of Bayesian networks have been developed for a wide variety of subject areas, in tasks such as learning, modeling, forecasting and decision-making. Out of hundreds of related papers found, we picked a sample of 150 to study the trends of such applications over a 16-year interval. We classified the publications according to their corresponding domain of application, and then analyzed the tendency to develop Bayesian networks in determined areas of research. We found a set of indicators that help better explain these tendencies: the levels of formalization, data accuracy and data accessibility of a domain, and the level of human intervention in the primary data. The results and methodology of the current study provide insight into potential areas of research and application of Bayesian networks.

Keywords: Bayesian networks · Uncertainty · Domain
Formalization · Human intervention · Data accuracy
Data accessibility

1 Background

Uncertainty is a constant in most aspects of everyday life, making it necessary to develop tools that can aid in minimizing it. Bayesian networks (BNs) [16] are nowadays a popular and important method for reasoning under conditions of uncertainty in artificial intelligence (AI). BNs are popular for modeling a wide variety of domains because they facilitate both the construction of the models and the understanding of the domain.

From the modeling point of view, a combination of empirical data and judgment from experts can be used to build BNs. This flexibility is a very useful feature that has attracted researchers from diverse fields. In addition, BNs can be represented as network graphs to provide a visualization of the components and dependencies of a subject area. From the knowledge point of view, they represent domain variables in a probabilistic way, allowing inference under uncertainty and making it possible to run a model with missing data. Therefore, due

A. Lupeikiene et al. (Eds.): DB&IS 2018, CCIS 838, pp. 376–390, 2018.
https://doi.org/10.1007/978-3-319-97571-9_29

to the characteristics of BNs as tools for modeling uncertainty, their possible areas of application can widely vary.

Surveys regarding the application of BNs in specific subject areas can be found nowadays (e.g. [13,14]). However, to the best of our knowledge, the development of BNs throughout multiple domains has not been a subject of research yet. This work aims at providing a general picture about the application of BNs along multiple subject areas, underlining the reasons that have made possible the development of these statistic tools in each domain. While being far from comprehensive, this study is expected to point at current trends in the application of BNs, as well as future potential areas for their development.

2 Introduction to Bayesian Networks

A Bayesian network (BN) [16] is a directed acyclic graph (DAG) in which random variables correspond to nodes and edges between nodes are conditional dependencies between the variables. These random variables reflect knowledge uncertainty in a subject area. If the variables are continuous, a common approach is to discretize them by dividing them into intervals. Thus, nodes are partitioned into a set of possible states that are able to represent numerical and non-numerical values.

An arc or edge from node X_i to node X_j represents, intuitively, that X_i has a direct effect or impact on X_j [19]. X_i is defined as a parent of X_j; thus, all those nodes that have arcs directed to a specific node are considered its parents. Each node in the network has a conditional probability distribution associated of the form $P(X_i|parents(X_i))$. This is represented as a conditional probability table (CPT), which contains all the parent influences that act upon the variable X_j. A joint probability distribution (JPD), which is the likelihood of each possible event as defined by the CPTs, can be calculated using the chain rule:

$$P(X_1, ..., X_n) = \prod_{i=1}^{n} P(X_i|Parents(X_i))$$

The process of calculation in a BN model is based on the Bayes theorem, where for two uncertain variables A and D:

$$P(A|D) = \frac{P(D|A)P(A)}{P(D)}$$

3 Subject of Study and Data Structure

A total of 9048 related papers published between the years 2000–2016 were found using the search engine for academic publications Google Scholar [6], filtering results by requiring the appearance of synonyms for the term “Bayesian network” in the title. The distribution of related publications found per year can be visualized in Fig. 1.

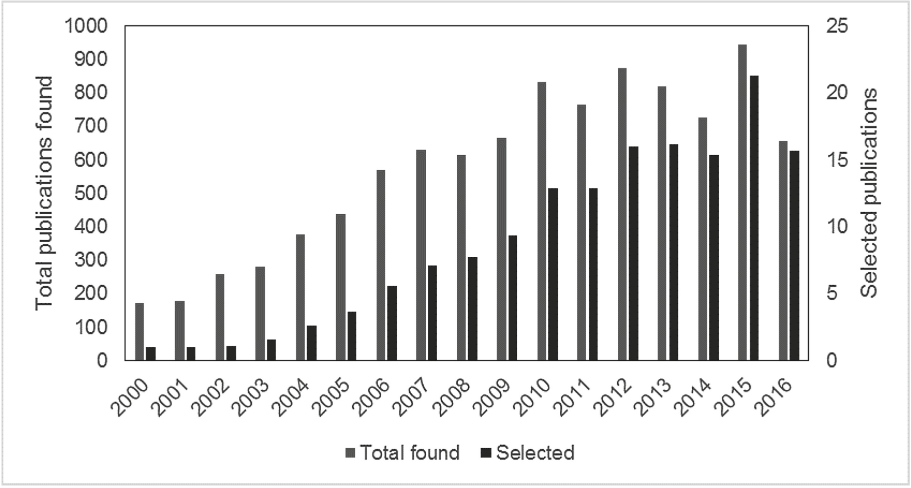


Fig. 1. Total number of related publications found vs selected number of papers (per year)

Due to limitations regarding time and resources, we randomly selected an initial sample of 422 articles from all the publications found. We excluded merely descriptive papers (without any real-world application of a BN), and thoroughly searched the remaining papers for properly presented BNs, i.e. articles that not only mention the utilization of a BN, but also present a detailed structure of the network (nodes, conditional dependencies, etc.). Considering these restrictions, our sample was reduced to 150 papers concerning real-world applications of BNs to multiple subject areas, distributed according to Fig. 1.

It is important to notice that the list of selected publications resulting of this process cannot be considered to be comprehensive or unbiased. For example, relevant publications may not have been detected using the current search terms, search results are strongly biased toward publications written in English, and books and theses were excluded from this review. Despite these caveats, it is likely that the publications in study provide a representative overview of the different areas of application of BNs.

The main question of the current investigation is, how is a BN related to a domain of research? We tackled this question from two angles. First, after examining each publication in detail, we extracted information concerning the applied BN: (i) the task trying to be solved by the BN, to gather knowledge about the source of uncertainty in the domain; (ii) the kind of variable represented by Bayesian nodes, to determine if the BN implementation requires human interaction; (iii) the stage of development of the BN, to determine if the subject area is relatively new or old; and (iv) the domain of application of the BN.

This last point connects the first approach with the second, which focuses on analyzing the domain of research in search for the answer. For this purpose, we characterized the domains by means of five criteria; namely, the dimension of the domain, its citation trend, and the levels of formalization, data accessibility and data accuracy. Observed differences in the application of BNs justified the inclusion of an extra characteristic: the level of human intervention. Regarding this analysis, judgment was often required to interpret the information provided in the papers, because many of the methods used were not fully described in the text. Therefore, it should be taken into account that the information presented here is based on an interpretation of what the authors presented.

4 Analysis of BN Publication Activity in Different Domains

The publications included in the review were grouped by subject area (domain). This classification was performed intuitively, taking into account factors as keywords in the text, the research institutions involved, and the publication journal. It should be noticed that the domain categories are not rigorously mutually exclusive: for example, artificial intelligence and informatics are considered as separate categories, but they can arguably overlap. Concretely, because of the usefulness of AI in a wide array of subject areas (BNs are considered part of it), we interpret the term AI as encompassing the tools and techniques not covered by other domains. For example, Multi-agent systems, Semantic Web and Conversation analysis, which can be considered as AI sub-domains, are classified separately due to their greater use of BNs in relation to other subject areas.

Table 1 shows the quantity of selected papers over 15 domains, along with the dimension of each domain, the averaged citations per year (per publication), and the citation trend as well. Citations per year is an indicator relative to a single paper; it is obtained by dividing the citation number of the paper by how many years passed from publication up until today. The average citations per year, multiplied by the total expected papers for a determined year, results in the total expected citation number in the domain for that period. Forecasts can be made by including the citation trend.

The trend indicator in Table 1 is the slope of the simple regression over the scatter plotting the sum of citations per year from 2000 to 2016. A negative trend indicates citation stagnation, while a positive trend indicates citation growth. The dimension is an indicator of the size of the domain: it is calculated as the area under the simple regression curve where the slope is the citation trend and the axis intersection is the sum of citations, and is afterwards normalized to take values between a minimum of 0 and a maximum of 1. A visual representation of Table 1 is presented in Fig. 2.

We consider that the indicators presented in Table 1 can give us a starting point for the analysis of each subject area in terms of growth, size and influence in the scientific community. However, it should be noted that conclusions based on these indicators must consider an important bias, which we will call compatibility bias. It refers to the presence or absence of likelihood between the characteristics of the subject area and the tasks commonly solved by means of BN. In other words, we cannot jump into conclusions about a domain as a whole, because these

Table 1. Publication and citation metrics per domain (QTY=Quantity; DIM=Dimension; CY=Citations/year; CT=Citation trend)

Subject area	SA	QTY	DIM	CY	CT
Artificial Intelligence	AI	23	0.58	18.68	−0.3
Biology	BIO	6	0.34	30.29	−0.35
Computer science	CS	9	0.39	17.22	0.23
Conversation analysis	CA	3	0.08	13.77	−0.11
Economics	ECO	6	0.14	13.48	−0.03
Environmental science	ES	21	0.95	19.88	−0.35
Genetics	GEN	23	1	55.73	−1
Industrial systems	IS	21	0.42	21.93	0.36
Informatics	INF	3	0.2	39.13	0.07
Law	LAW	3	0.08	16.75	0.16
Medicine	MED	6	0.49	34	0.68
Multi-agent systems	MAS	6	0.22	14.81	−0.2
Networks	NET	6	0.15	15.44	0.05
Semantic Web	SW	3	0.07	12.67	−0.07
Transportation	TRA	11	0.42	17.32	0.31

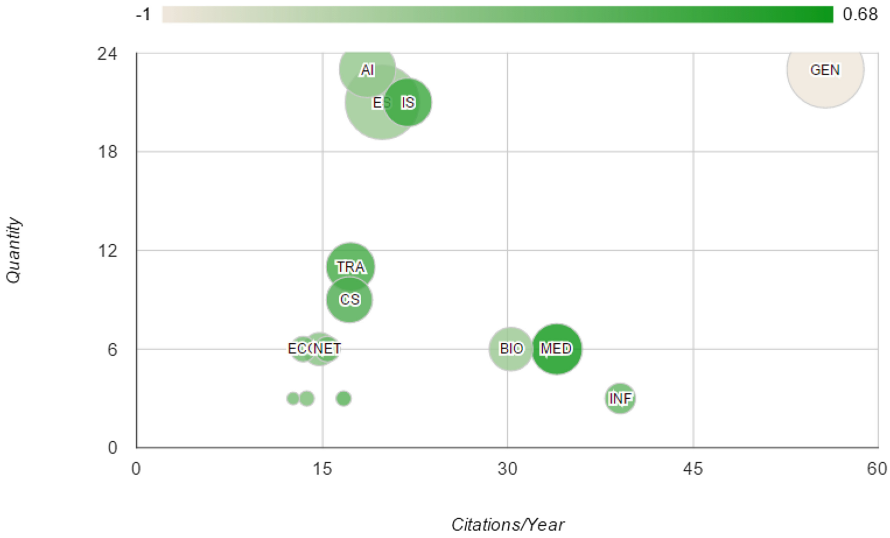


Fig. 2. Publication and citation metrics per domain. The color scale corresponds to the citation trend: a light color indicates a negative trend, while a dark color indicates citation growth. The circle diameter corresponds to the domain dimension.

indicators only reveal information about a subject area from the point of view of BN applications. Nevertheless, the scientific status of the current subject areas can be estimated without considering the compatibility bias (for the moment).

Independent from the number of citations, high number of publications may relate to a well-consolidated domain, whereas low number of publications may point to an undeveloped, possibly new scientific domain. High number of citations suggests considerable scientific interest in the field, while low number of citations indicate either lack of scientific interest or the generation of new, previously nonexistent knowledge.

A high dimension and citation trend suggests that the domain is on the rise with constant new research, but at the same time is well-consolidated throughout the last years. This can be seen in the case of artificial intelligence, industrial systems, transport, computer science and medicine. Alternatively, a low dimension and trend, like in biology and economics, points to a stagnation in the domain. A positive citation trend with a low dimension is most probably a new subject area with considerable scientific potential, as in law, informatics and networks.

A different dynamic is noticed in genetics, where the dimension is considerably higher than the citation trend. In this case, two possible explanations can be derived. On the one hand, the domain could already be well-established but research has been halted or closed, possibly by a comprehensive translation of theoretical models to practical appliances. On the other hand, the domain could be recently discovered and the information space is wide but unexplored, giving researchers the opportunity to generate new knowledge from different focal points without the need to rely on citations to previous work.

Besides their inherent probabilistic modeling function, the reviewed BNs in this study serve different tasks or purposes. Figure 3 presents an overview of the domains with the addition of the tasks that are solved by BNs.

It can be noticed in Fig. 3 that the task of learning through BNs is present only in genetics and biology, underlining the knowledge-gathering character of these fields. The tasks of learning, classification and recommendation don't make use of human decision-making, while the rest vary in relation to its utilization.

Different stages of development were observed in the reviewed BNs. Structure learning is present in 15.4% of the reviewed papers, mostly in the field of genetics. This suggests an abundance of primary data in that domain, together with an absence of existing knowledge from where to build the models. Parameter learning is reported in the areas of environmental science, biology, genetics, medicine and artificial intelligence, comprising 13.5% of the papers. Bayesian network models ready up to the inference stage represent the remaining 71.2%, where 44.2% are validated and the remaining 26.9% are not.

5 Impact of Domain Properties on the Application of BNs

As we already mentioned, we consider that the indicators already presented do not represent a transparent picture of the "growth" or "progress" of a domain.

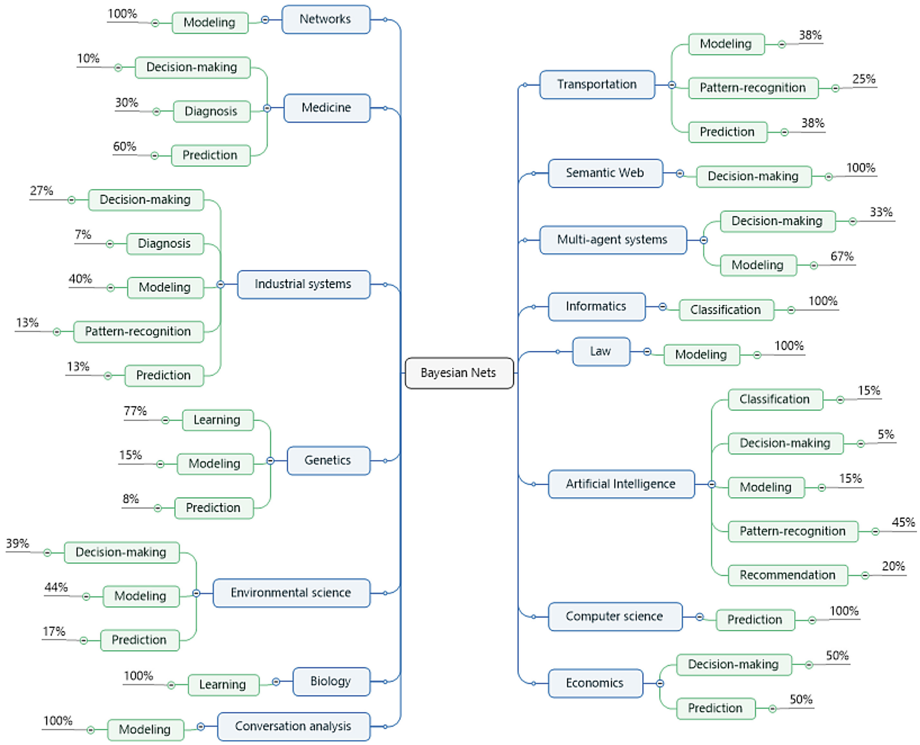


Fig. 3. Tasks addressed by BNs for each subject area

Instead, they portray an inclination to develop BN applications in a determined area of research. This inclination can be described as a compatibility bias, or the likelihood of applying BNs in a domain that facilitates their implementation. In order to analyze this bias, we firstly hypothesized that there are specific attributes of a domain that determine its suitability for BN applications; namely, levels of formalization, data accessibility and data accuracy.

We use the formalization indicator as the level to which sets of symbols, formulas and rules are used to describe objects, events and their interrelationships in the domain. Accordingly, the level of formalization is defined by the presence or absence of standards, mathematical formulas, languages, or any kind of unified coding of the primary data of a domain.

Data accessibility is the level to which it is possible for an independent researcher to obtain data from primary sources in the domain. Monopolization of information in a domain, for example, is considered as a negative factor regarding accessibility to information. Open access to primary data or the need for specialized equipment to obtain it also affect this indicator.

Data accuracy is the unlikelihood of finding contradictions, ambiguity and noise in the information gathered from the domain sources. In a domain with low data accuracy, for example, the researcher knows which information to retrieve from the primary source but is unable to fully obtain it due to unavoidable error in measurement or interpretation. It represents the gap between the real data and the one gathered by the researcher.

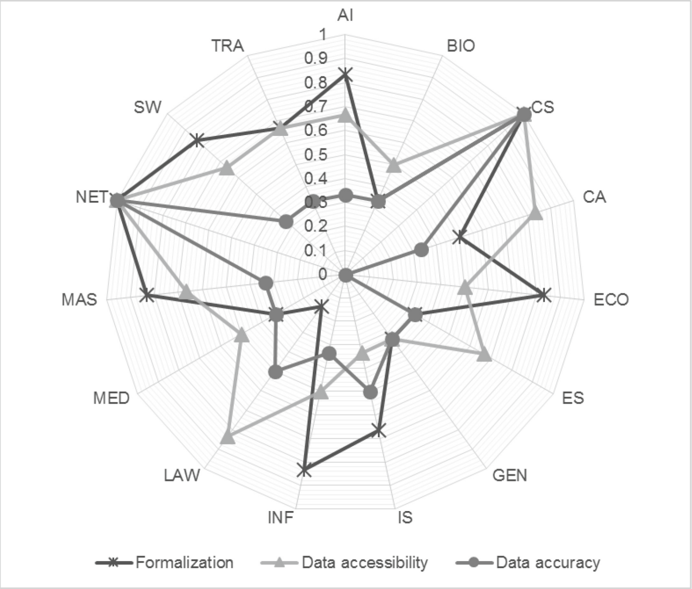


Fig. 4. Formalization, data accessibility and data accuracy in different domains

Figure 4 presents an attempt to quantify the mentioned attributes in the domains of the present study. It is necessary to underline that arguments found in the literature were used (when possible) to obtain a qualitative score for these indicators and subjective judgments were needed in the rest of the cases.

5.1 Formalization

The areas of computer science and networks are rated with the highest level of formalization because the languages in which they are expressed were created for themselves, not trying to imitate or model behaviors outside their subject area. However, formalization does not mean universality, which means that these fields can be expressed in a variety of languages and symbolizations that can express the same concepts, possibly with mappings between each other.

The next level of formalization comprises fields as artificial intelligence, informatics and industrial systems. Although artificial intelligence shares aspects with computer science, the difficulty in formalizing lies in the attempt to model human intelligence. As an example, [12] raises the issue that no formal theory of common sense can get by without some formalization of context. Informatics is less formalized than computer science because it involves a broader arrange of subfields, each of which tries to apply established formalizations of computer science to itself. Economics relies heavily in mathematical and statistical formalizations, to the extent that there is influence to diminish this trend [9].

The fields of transportation and industrial systems possess a medium-high level of formalization. Transportation problems in operations research are tightly related to mathematical optimization, and dedicated formalization attempts have been developed [7]. Concerning industrial systems, there is a substantial number of international standards regarding manufacturing and other industrial systems, in which common languages are established among specific disciplines. Formalization efforts of a medium level have been made in the field of conversation analysis [8], however not widespread.

A low-medium level of formalization was assigned to the fields of biology, environmental science, genetics and medicine, as mathematical formalization of the concepts on the processes in living systems represents considerable difficulties [18]. Environmental science is tightly related to public policy, therefore formalization attempts are in early stages [10]. The domain of law is behind the rest of fields with a low formalization level. However, progress has been made in the field of argumentation [3,17].

5.2 Data Accessibility

The areas of computer science and networks are rated once again with the highest level, in this case, of data accessibility. This is because the body of related data is artificial, constantly updated and open for research and application. Both conversation analysis and law share a high level of data availability because audio-visual media, a source of conversational interactions, is widespread and freely available in the Internet, while the body of the law is in the public domain for public access.

Environmental science, artificial intelligence and transportation were rated with a medium-high level of data availability. Environmental data is in the public domain, without commercial restrictions. However, only recently has it become crucial in research with issues like climate change. Artificial intelligence, although is a wide field with considerable quantity of information sources, critical data is kept outside of the public domain, as it is a commercial competitive advantage (e.g. Google, Microsoft). Transportation shares similar characteristics with artificial intelligence, in this respect.

Medicine, biology, economics and informatics share a medium level of data accessibility. In medicine, research results are openly available and constantly scrutinized by governmental authorities. However, primary data is not publicly available in as much as two thirds of the research performed [1]. In biology, independent researchers are still able to gather information possibly without the need for specialized technology. In economics, a considerable percentage of data is publicly available, but it can be argued that its veracity depends on hidden factors (e.g. political). In informatics, knowledge management capabilities are significantly related with competitive advantage [2], which is a constraint to data openness.

Genetics and industrial systems were rated with a low-medium score on data availability. Information generation in genetics depends on private and governmental funding in specialized laboratories, under specialized research programs. Concerning industrial systems, data is generally not publicly open, especially in industry areas high in competitive advantage.

5.3 Data Accuracy

The area of economics possesses a very low level of data accuracy, because there is inherent uncertainty in dealing with out of control, external behavior in a big scale market involving a mass of agents. Industrial systems and law are situated in a medium level of uncertainty. Data accuracy of the law is no less than moderate and it is a much less serious defect in the law than it is often thought to be [11]. In relation to industrial systems, uncertainty is present in the creation, operation and control of industrial processes.

Data accuracy in computer science and networks is virtually the highest, resting on the fact that methods are constantly developed to deal with uncertainty in a fully artificial domain. The rest of subject areas were rated with a low-medium level of data accuracy. In medicine, biology and genetics, there is significant doubt on the prospects for highly deterministic and basically similar mechanisms between individuals [5]. In environmental science, the interconnectedness of complex systems keeps a considerable level of uncertainty. In fact, representing uncertainty in environmental policy is an important subject of study [20].

6 Impact of Primary Data Human Intervention on the Application of BNs

During the course of this study, however, we noticed that not only the attributes of a subject area as a whole can determine its suitability for BN applications, but also the characteristics of the primary data source (we consider primary data as the concepts represented by the BN variables and their respective states in each reviewed publication). For this reason we introduce an additional indicator, the human factor, which refers to the level in which human intervention, and conversely, external factors, change or alter the primary data of the domain.

The indicator can take three levels: −1, meaning that there is no decision-making or human control influencing the domain data; 0, which means that there is a perceptible level of human influence on the data; and 1, when practically all the domain data is product of human actions and interactions. A thorough revision of the Bayesian variables in each selected model was the basis for the human factor scoring (see Table 2). A classification of the subject areas by this factor is presented in Fig. 5.

The areas of artificial intelligence, biology, genetics, informatics and law are devoid of any perceptible human factor in the application of Bayesian networks,

Table 2. Distribution of publications (NP) according to Bayesian variables, domain (SA) and human factor (HF)

Bayesian variables	SA	NP	HF
Accident factors	IS	6	0
Accident factors	TRA	3	0
Acoustic-linguistic patterns	AI	2	−1
Audio-visual patterns, previous audio-visual patterns	AI	3	−1
Bankruptcy factors	ECO	3	0
Biological variables	BIO	6	−1
Body part poses, spatial-visual factors	AI	3	−1
Capture factors	ES	2	0
Classes	AI	3	−1
Component wear, cutting parameters	IS	3	1
Concepts	SW	3	−1
Defect proneness, software metrics	CS	3	1
Defect resolution process stages	CS	3	1
Disease progression, treatment	MED	3	0
Economic impact factors	ECO	3	0
Ecosystem health indicators, biochemical properties	ES	2	0
Environmental drivers, ecological responses, decision costs and benefits	ES	3	0
Environmental impact factors	ES	2	0
Environmental viability factors	ES	3	0
Exploits	NET	6	1
Fish population impact factors	ES	3	0
Genes	GEN	20	−1
Hypothesis, Evidence	LAW	3	−1
Machining parameters	IS	3	1
Physical scenario variables	TRA	3	0
Physical symptoms	MED	3	0
Pollution factors	MAS	3	0
Population stressor factors	ES	3	0
Protein features	GEN	3	−1
Risk factor	IS	3	0
Road link vehicle flow	TRA	2	1
Rules, actions	AI	3	−1
Software quality metrics	CS	3	1
Structural, durational, lexical agreement features	CA	3	1

(*continued*)

Table 2. (*continued*)

Bayesian variables	SA	NP	HF
Sustainability factors	ES	3	0
System component status	IS	3	1
System components	IS	3	1
Traffic conditions	TRA	3	1
Trust factors	MAS	3	−1
User/actual experience	INF	3	−1
Users, items, features	AI	3	−1
Vehicle class features	AI	6	−1

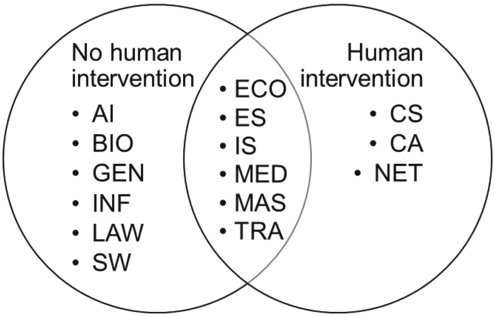


Fig. 5. Domain classification by human factor, based on the reviewed publications. The overlap indicates partial human intervention.

with a level of −1. Arguably, law is a field with an inherent human factor. However, the subfield of argumentation [3] deals with the logic of argumentation instead of the legal confrontations themselves.

Computer science, conversation analysis and networks are fully artificial domains in the samples gathered for the present study, with a human factor level of 1. Software defects [15], network exploits [21] and conversation agreement features [4] depend on the subjective, conscious or unconscious, behavior of humans for the existence of the corresponding sub-domains. The remaining subject areas correspond to a mix of subjective human influence and objective observation of external factors in the domain. For example, in our sample, the highly formalized field of economics presents an undetectable human factor, but a factor of 0 is assigned due to its dependence on human-determined parameters.

7 Discussion

Instead of representing different domains, Fig. 6 distributes the totality of publications into the spectrum of data accuracy, accessibility, formalization and

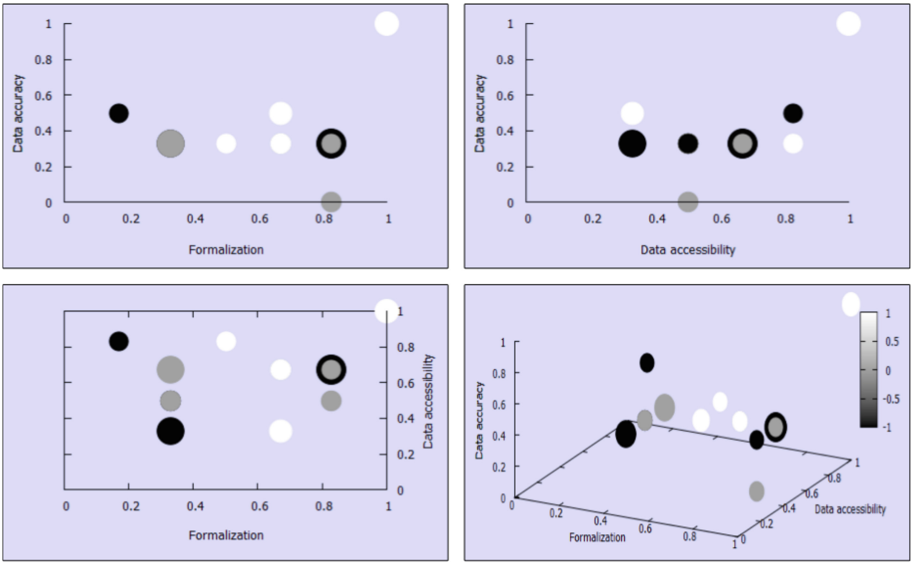


Fig. 6. Distribution of the number of publications (size of the spheres) according to formalization, data accuracy, data accessibility and human factor (color scale, dark being negative and white positive).

human factor. It shows that BN applications fully involving human intervention are roughly associated to high levels of formalization, high levels of data accuracy and varied levels of data accessibility. Examples are computer science, conversation analysis and networks. Applications with a negative human factor (not involving human intervention) are related to low levels of data accuracy, varied levels of formalization but at the same time high levels of data accessibility. Examples are artificial intelligence, biology, genetics, informatics, and law. The rest of BN applications have a mixed level of human intervention and are related to low levels of data accuracy, medium levels of data accessibility, and varied levels of formalization.

Domains that present a high citation dimension and a low citation trend, such as environmental science and AI, are considered suitable for BN development and are expected to present a stable utilization of such networks in the near future. On the other hand, it was found that the tasks of learning, classification and recommendation are associated only to a negative human factor. It can be argued that these tasks reflect an early development of the subject area: for example, the domain of genetics is dominated by learning tasks and BN structure learning applications. Thus, a positive citation trend in this and other domains suggests that publications involving human intervention in the primary data can be expected in the near future, along with new decision-making tasks.

8 Conclusions

This paper has presented a qualitative and quantitative analysis of the application of BNs in different subject areas. Common indicators such as number of publications reviewed and their citations were used to create more general indicators for each subject area, like the dimension and citation trend of a domain. The purpose of these indicators was not to represent “growth” or “progress” of a domain of research. Instead, they portray a compatibility bias, or an inclination to develop BN applications in a determined area of research, mainly because of the suitability of BNs.

Our strategy to quantify this compatibility bias was to introduce three additional criteria for domain analysis; namely, levels of formalization, data accessibility and data accuracy. The final step was to verify if these three criteria are suitable to explain the trends of BN applications in each field. At this point, we found that also the characteristics of the primary data source must be taken into account. Therefore, an additional factor was introduced into the analysis: the human intervention factor.

The analysis of the four resulting indicators gave us a list of conclusions.

- Full human intervention is associated to high levels of formalization, high levels of data accuracy and varied levels of data accessibility.
- Absence of human intervention is related to low levels of data accuracy, varied levels of formalization but at the same time high levels of data accessibility.
- Mixed level of human intervention are related to low levels of data accuracy, medium levels of data accessibility, and varied levels of formalization.

We expect that domains that comprise the mentioned combinations of formalization, data accessibility, data accuracy and human intervention will be suitable for the development of BN applications in the future. These indicators are meant to facilitate the analysis and development of BN applications. The dimension and citation trends that we presented provide the current trends in developing BNs, but also give possible research opportunities: by applying the present methodology in subject areas not present in this study, new possibilities for BN development can be found.

Acknowledgment. The research presented in this paper was supported by the RUDN University Program 5-100.

References

1. Alsheikh-Ali, A.A., Qureshi, W., Al-Mallah, M.H., Ioannidis, J.P.A.: Public availability of published research data in high-impact journals. PLoS ONE **6**(9), e24357 (2011). https://doi.org/10.1371/journal.pone.0024357
2. Chuang, S.H.: A resource-based perspective on knowledge management capability and competitive advantage: an empirical investigation. Expert Syst. Appl. **27**(3), 459–465 (2004). https://doi.org/10.1016/j.eswa.2004.05.008

3. Fenton, N., Neil, M., Lagnado, D.A.: A general structure for legal arguments about evidence using Bayesian networks. Cogn. Sci. **37**(1), 61–102 (2012). https://doi.org/10.1111/cogs.12004
4. Galley, M., McKeown, K., Hirschberg, J., Shriberg, E.: Identifying agreement and disagreement in conversational speech: Use of Bayesian networks to model pragmatic dependencies, pp. 669–676 (2004)
5. Greenspan, R.J.: Biological indeterminacy. Sci. Eng. Ethics **18**(3), 447–452 (2012). https://doi.org/10.1007/s11948-012-9379-2
6. Harzing, A., van der Wal, R.: Google scholar as a new source for citation analysis. Ethics Sci. Environ. Polit. **8**, 61–73 (2008). https://doi.org/10.3354/esep00076
7. Xu, S.J., Nourinejad, M., Lai, X., Chow, Y. J.: Network learning via multiagent inverse transportation problems. Transp. Sci. (2017). https://doi.org/10.1287/trsc.2017.0805
8. Kasper, G., Wagner, J.: Conversation analysis in applied linguistics. Ann. Rev. Appl. Linguist. **34**, 171–212 (2014). https://doi.org/10.1017/S0267190514000014
9. Katzner, D.W.: Unmeasured Information and the Methodology of Social Scientific Inquiry, 1st edn. Kluwer Academic Publishers, Boston (2001)
10. van Kerkhoff, L.: Integrated research: concepts of connection in environmental science and policy. Environ. Sci. Policy **8**(5), 452–463 (2005). https://doi.org/10.1016/j.envsci.2005.06.002
11. Kress, K.: Legal indeterminacy. Calif. Law Rev. **77**(2), 283 (1989). https://doi.org/10.2307/3480606
12. McCarthy, J.: Generality in artificial intelligence. Commun. ACM **30**(12), 1030–1035 (1987). https://doi.org/10.1145/33447.33448
13. Mkrtchyan, L., Podofillini, L., Dang, V.: Bayesian belief networks for human reliability analysis: a review of applications and gaps (2015)
14. Newton, A.C.: Bayesian Belief Networks in Environmental Modelling: A Review of Recent Progress, 1st edn, pp. 13–50. Nova Science Publishers, New York (2009)
15. Okutan, A., Yildiz, O.T.: Software defect prediction using Bayesian networks. Empir. Softw. Eng. **19**(1), 154–181 (2012). https://doi.org/10.1007/s10664-012-9218-8
16. Pearl, J.: Probabilistic Reasoning in Intelligent Systems, 1st edn. Kaufmann, San Mateo (1988)
17. Prakken, H.: A logical framework for modelling legal argument, pp. 1–9. ACM Press (1993)
18. Rubin, A., Riznichenko, G.Y.: Mathematical Biophysics, 1st edn. Springer, Boston (2014). https://doi.org/10.1007/978-1-4614-8702-9
19. Russell, S.J., Norvig, P., Davis, E.: Artificial Intelligence, 1st edn. Prentice Hall, Upper Saddle River (2010)
20. Shackley, S., Wynne, B.: Representing uncertainty in global climate change science and policy: boundary-ordering devices and authority. Sci. Technol. Hum. Values **21**(3), 275–302 (1996). https://doi.org/10.1177/016224399602100302
21. Xie, P., Li, J.H., Ou, X., Liu, P., Levy, R.: Using Bayesian networks for cyber security analysis, pp. 211–220. IEEE/IFIP (2010)

Author Index