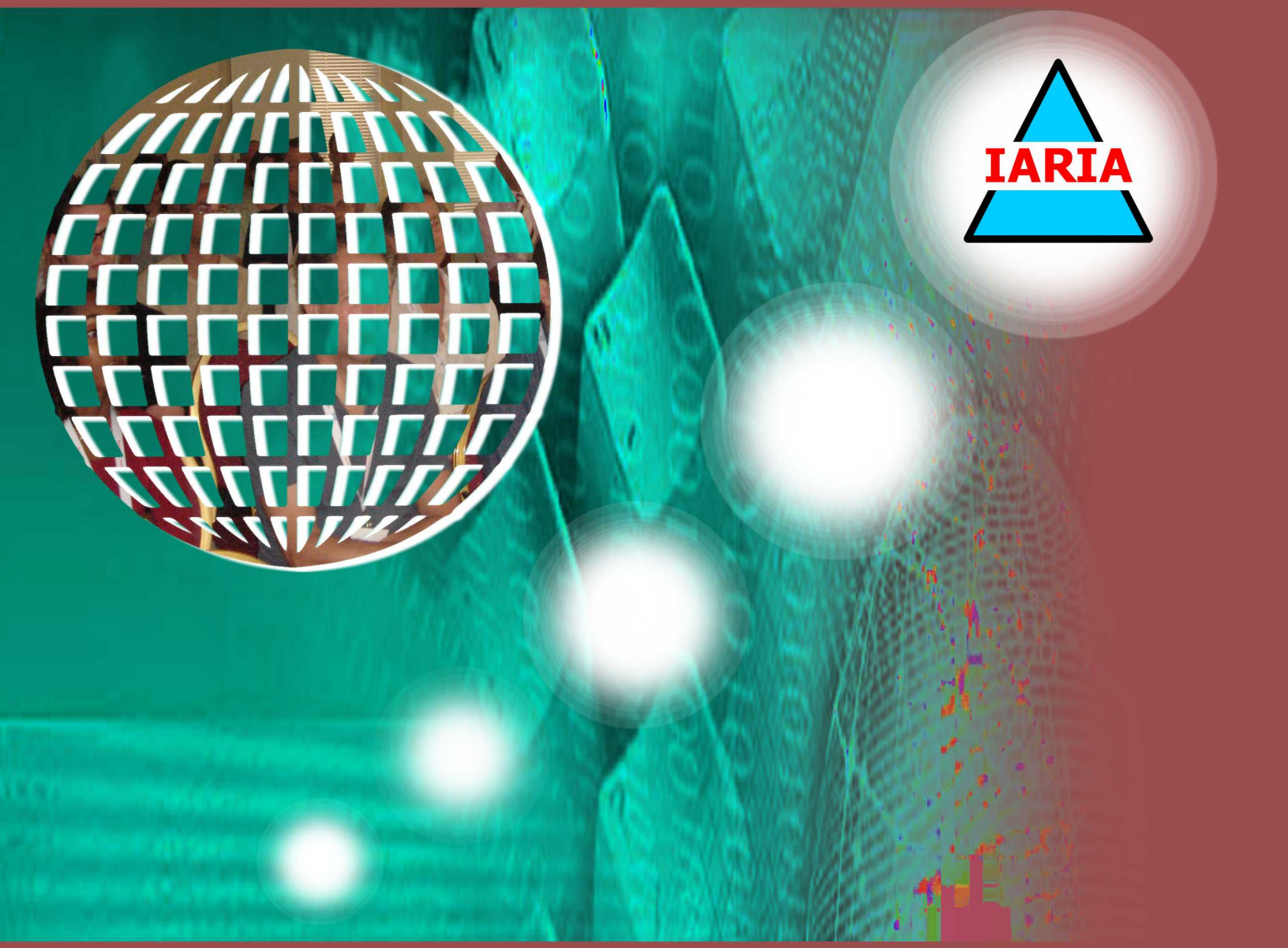
International Journal on Advances in Software



2014 vol. 7 nr. 3&4

The *International Journal on Advances in Software* is published by IARIA.
ISSN: 1942-2628
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Software
Volume 7, Number 3 & 4, 2014

Farid Bourennani, University of Ontario Institute of Technology (UOIT), Canada
Narhimene Boustia, Saad Dahlab University - Blida, Algeria
Hongyu Pei Breivold, ABB Corporate Research, Sweden
Carsten Brockmann, Universität Potsdam, Germany
Antonio Bucchiarone, Fondazione Bruno Kessler, Italy
Georg Buchgeher, Software Competence Center Hagenberg GmbH, Austria
Dumitru Burdescu, University of Craiova, Romania
Martine Cadot, University of Nancy / LORIA, France
Isabel Candal-Vicente, Universidad del Este, Puerto Rico
Juan-Vicente Capella-Hernández, Universitat Politècnica de València, Spain
Jose Carlos Metrolho, Polytechnic Institute of Castelo Branco, Portugal
Alain Casali, Aix-Marseille University, France
Yaser Chaaban, Leibniz University of Hanover, Germany
Savvas A. Chatzichristofis, Democritus University of Thrace, Greece
Antonin Chazalet, Orange, France
Jiann-Liang Chen, National Dong Hwa University, China
Shiping Chen, CSIRO ICT Centre, Australia
Wen-Shiung Chen, National Chi Nan University, Taiwan
Zhe Chen, College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, China PR
Po-Hsun Cheng, National Kaohsiung Normal University, Taiwan
Yoonsik Cheon, The University of Texas at El Paso, USA
Lau Cheuk Lung, INE/UFSC, Brazil
Robert Chew, Lien Centre for Social Innovation, Singapore
Andrew Connor, Auckland University of Technology, New Zealand
Rebeca Cortázar, University of Deusto, Spain
Noël Crespi, Institut Telecom, Telecom SudParis, France
Carlos E. Cuesta, Rey Juan Carlos University, Spain
Duilio Curcio, University of Calabria, Italy
Mirela Danubianu, "Stefan cel Mare" University of Suceava, Romania
Paulo Asterio de Castro Guerra, Tapijara Programação de Sistemas Ltda. - Lambari, Brazil
Cláudio de Souza Baptista, University of Campina Grande, Brazil
Maria del Pilar Angeles, Universidad Nacional Autonónoma de México, México
Rafael del Vado Vírseda, Universidad Complutense de Madrid, Spain
Giovanni Denaro, University of Milano-Bicocca, Italy
Hepu Deng, RMIT University, Australia
Nirmit Desai, IBM Research, India
Vincenzo Deufemia, Università di Salerno, Italy
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Javier Diaz, Rutgers University, USA
Nicholas John Dingle, University of Manchester, UK
Roland Dodd, CQUniversity, Australia
Aijuan Dong, Hood College, USA
Suzana Dragicevic, Simon Fraser University- Burnaby, Canada
Cédric du Mouza, CNAM, France
Ann Dunkin, Palo Alto Unified School District, USA

Jana Dvorakova, Comenius University, Slovakia
Lars Ebrecht, German Aerospace Center (DLR), Germany
Hans-Dieter Ehrich, Technische Universität Braunschweig, Germany
Jorge Ejarque, Barcelona Supercomputing Center, Spain
Atilla Elçi, Aksaray University, Turkey
Khaled El-Fakih, American University of Sharjah, UAE
Gledson Elias, Federal University of Paraíba, Brazil
Sameh Elnikety, Microsoft Research, USA
Fausto Fasano, University of Molise, Italy
Michael Felderer, University of Innsbruck, Austria
João M. Fernandes, Universidade de Minho, Portugal
Luis Fernandez-Sanz, University of de Alcala, Spain
Felipe Ferraz, C.E.S.A.R, Brazil
Adina Magda Florea, University "Politehnica" of Bucharest, Romania
Wolfgang Fohl, Hamburg Universiy, Germany
Simon Fong, University of Macau, Macau SAR
Gianluca Franchino, Scuola Superiore Sant'Anna, Pisa, Italy
Naoki Fukuta, Shizuoka University, Japan
Martin Gaedke, Chemnitz University of Technology, Germany
Félix J. García Clemente, University of Murcia, Spain
José García-Fanjul, University of Oviedo, Spain
Felipe Garcia-Sanchez, Universidad Politecnica de Cartagena (UPCT), Spain
Michael Gebhart, Gebhart Quality Analysis (QA) 82, Germany
Tejas R. Gandhi, Virtua Health-Marlton, USA
Andrea Giachetti, Università degli Studi di Verona, Italy
Robert L. Glass, Griffith University, Australia
Afzal Godil, National Institute of Standards and Technology, USA
Luis Gomes, Universidade Nova Lisboa, Portugal
Diego Gonzalez Aguilera, University of Salamanca - Avila, Spain
Pascual Gonzalez, University of Castilla-La Mancha, Spain
Björn Gottfried, University of Bremen, Germany
Victor Govindaswamy, Texas A&M University, USA
Gregor Grambow, University of Ulm, Germany
Carlos Granell, European Commission / Joint Research Centre, Italy
Christoph Grimm, University of Kaiserslautern, Austria
Michael Grottke, University of Erlangen-Nuernberg, Germany
Vic Grout, Glyndwr University, UK
Ensar Gul, Marmara University, Turkey
Richard Gunstone, Bournemouth University, UK
Zhensheng Guo, Siemens AG, Germany
Phuong H. Ha, University of Tromso, Norway
Ismail Hababeh, German Jordanian University, Jordan
Shahliza Abd Halim, Lecturer in Universiti Teknologi Malaysia, Malaysia
Herman Hartmann, University of Groningen, The Netherlands
Jameleddine Hassine, King Fahd University of Petroleum & Mineral (KFUPM), Saudi Arabia
Tzung-Pei Hong, National University of Kaohsiung, Taiwan

Peizhao Hu, NICTA, Australia
Chih-Cheng Hung, Southern Polytechnic State University, USA
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Noraini Ibrahim, Universiti Teknologi Malaysia, Malaysia
Anca Daniela Ionita, University "POLITEHNICA" of Bucharest, Romania
Chris Ireland, Open University, UK
Kyoko Iwasawa, Takushoku University - Tokyo, Japan
Mehrshid Javanbakht, Azad University - Tehran, Iran
Wassim Jaziri, ISIM Sfax, Tunisia
Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Jinyuan Jia, Tongji University. Shanghai, China
Maria Joao Ferreira, Universidade Portucalense, Portugal
Ahmed Kamel, Concordia College, Moorhead, Minnesota, USA
Teemu Kanstrén, VTT Technical Research Centre of Finland, Finland
Nittaya Kerdprasop, Suranaree University of Technology, Thailand
Ayad ali Keshlaf, Newcastle University, UK
Nhien An Le Khac, University College Dublin, Ireland
Sadegh Kharazmi, RMIT University - Melbourne, Australia
Kyoung-Sook Kim, National Institute of Information and Communications Technology, Japan
Youngjae Kim, Oak Ridge National Laboratory, USA
Roger "Buzz" King, University of Colorado at Boulder, USA
Cornel Klein, Siemens AG, Germany
Alexander Knapp, University of Augsburg, Germany
Radek Koci, Brno University of Technology, Czech Republic
Christian Kop, University of Klagenfurt, Austria
Michal Krátký, VŠB - Technical University of Ostrava, Czech Republic
Narayanan Kulathuramaiyer, Universiti Malaysia Sarawak, Malaysia
Satoshi Kurihara, Osaka University, Japan
Eugenijus Kurilovas, Vilnius University, Lithuania
Philippe Lahire, Université de Nice Sophia-Antipolis, France
Alla Lake, Linfo Systems, LLC, USA
Fritz Laux, Reutlingen University, Germany
Luigi Lavazza, Università dell'Insubria, Italy
Fábio Luiz Leite Júnior, Universidade Estadual da Paraiba,Brazil
Alain Lelu, University of Franche-Comté / LORIA, France
Cynthia Y. Lester, Georgia Perimeter College, USA
Clement Leung, Hong Kong Baptist University, Hong Kong
Weidong Li, University of Connecticut, USA
Corrado Loglisci, University of Bari, Italy
Francesco Longo, University of Calabria, Italy
Sérgio F. Lopes, University of Minho, Portugal
Pericles Loucopoulos, Loughborough University, UK
Alen Lovrencic, University of Zagreb, Croatia
Qifeng Lu, MacroSys, LLC, USA
Xun Luo, Qualcomm Inc., USA
Shuai Ma, Beihang University, China

Stephane Maag, Telecom SudParis, France
Ricardo J. Machado, University of Minho, Portugal
Maryam Tayefeh Mahmoudi, Research Institute for ICT, Iran
Nicos Malevris, Athens University of Economics and Business, Greece
Herwig Mannaert, University of Antwerp, Belgium
José Manuel Molina López, Universidad Carlos III de Madrid, Spain
Francesco Marcelloni, University of Pisa, Italy
Eda Marchetti, Consiglio Nazionale delle Ricerche (CNR), Italy
Leonardo Mariani, University of Milano Bicocca, Italy
Gerasimos Marketos, University of Piraeus, Greece
Abel Marrero, Bombardier Transportation, Germany
Adriana Martin, Universidad Nacional de la Patagonia Austral / Universidad Nacional del Comahue, Argentina
Goran Martinovic, J.J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
Stephan Mäs, Technical University of Dresden, Germany
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Jose Merseguer, Universidad de Zaragoza, Spain
Seyedeh Leili Mirtaheri, Iran University of Science & Technology, Iran
Lars Moench, University of Hagen, Germany
Yasuhiko Morimoto, Hiroshima University, Japan
Antonio Navarro Martín, Universidad Complutense de Madrid, Spain
Filippo Neri, University of Naples, Italy
Muaz A. Niazi, Bahria University, Islamabad, Pakistan
Natalja Nikitina, KTH Royal Institute of Technology, Sweden
Roy Oberhauser, Aalen University, Germany
Pablo Oliveira Antonino, Fraunhofer IESE, Germany
Rocco Oliveto, University of Molise, Italy
Sascha Opletal, Universität Stuttgart, Germany
Flavio Oquendo, European University of Brittany/IRISA-UBS, France
Claus Pahl, Dublin City University, Ireland
Marcos Palacios, University of Oviedo, Spain
Constantin Paleologu, University Politehnica of Bucharest, Romania
Kai Pan, UNC Charlotte, USA
Yiannis Papadopoulos, University of Hull, UK
Andreas Papasalouros, University of the Aegean, Greece
Rodrigo Paredes, Universidad de Talca, Chile
Päivi Parviainen, VTT Technical Research Centre, Finland
João Pascoal Faria, Faculty of Engineering of University of Porto / INESC TEC, Portugal
Fabrizio Pastore, University of Milano - Bicocca, Italy
Kunal Patel, Ingenuity Systems, USA
Óscar Pereira, Instituto de Telecomunicacoes - University of Aveiro, Portugal
Willy Picard, Poznań University of Economics, Poland
Jose R. Pires Manso, University of Beira Interior, Portugal
Sören Pirk, Universität Konstanz, Germany
Meikel Poess, Oracle Corporation, USA
Thomas E. Potok, Oak Ridge National Laboratory, USA

Christian Prehofer, Fraunhofer-Einrichtung für Systeme der Kommunikationstechnik ESK, Germany
Ela Pustułka-Hunt, Bundesamt für Statistik, Neuchâtel, Switzerland
Mengyu Qiao, South Dakota School of Mines and Technology, USA
Kornelije Rabuzin, University of Zagreb, Croatia
J. Javier Rainer Granados, Universidad Politécnica de Madrid, Spain
Muthu Ramachandran, Leeds Metropolitan University, UK
Thurasamy Ramayah, Universiti Sains Malaysia, Malaysia
Prakash Ranganathan, University of North Dakota, USA
José Raúl Romero, University of Córdoba, Spain
Henrique Rebêlo, Federal University of Pernambuco, Brazil
Hassan Reza, UND Aerospace, USA
Elvinia Riccobene, Università degli Studi di Milano, Italy
Daniel Riesco, Universidad Nacional de San Luis, Argentina
Mathieu Roche, LIRMM / CNRS / Univ. Montpellier 2, France
José Rouillard, University of Lille, France
Siegfried Rouvrais, TELECOM Bretagne, France
Claus-Peter Rückemann, Leibniz Universität Hannover / Westfälische Wilhelms-Universität Münster / North-German Supercomputing Alliance, Germany
Djamel Sadok, Universidade Federal de Pernambuco, Brazil
Ismael Sanz, Universitat Jaume I, Spain
M. Saravanan, Ericsson India Pvt. Ltd -Tamil Nadu, India
Idrissa Sarr, University of Cheikh Anta Diop, Dakar, Senegal / University of Quebec, Canada
Patrizia Scandurra, University of Bergamo, Italy
Giuseppe Scanniello, Università degli Studi della Basilicata, Italy
Daniel Schall, Vienna University of Technology, Austria
Rainer Schmidt, Munich University of Applied Sciences, Germany
Cristina Seceleanu, Mälardalen University, Sweden
Sebastian Senge, TU Dortmund, Germany
Isabel Seruca, Universidade Portucalense - Porto, Portugal
Kewei Sha, Oklahoma City University, USA
Simeon Simoff, University of Western Sydney, Australia
Jacques Simonin, Institut Telecom / Telecom Bretagne, France
Cosmin Stoica Spahiu, University of Craiova, Romania
George Spanoudakis, City University London, UK
Alin Stefanescu, University of Pitesti, Romania
Lena Strömbäck, SMHI, Sweden
Osamu Takaki, Japan Advanced Institute of Science and Technology, Japan
Antonio J. Tallón-Ballesteros, University of Seville, Spain
Wasif Tanveer, University of Engineering & Technology - Lahore, Pakistan
Ergin Tari, Istanbul Technical University, Turkey
Steffen Thiel, Furtwangen University of Applied Sciences, Germany
Jean-Claude Thill, Univ. of North Carolina at Charlotte, USA
Pierre Tiako, Langston University, USA
Božo Tomas, HT Mostar, Bosnia and Herzegovina
Davide Tosi, Università degli Studi dell'Insubria, Italy
Guglielmo Trentin, National Research Council, Italy

Dragos Truscan, Åbo Akademi University, Finland
Chrisa Tsinaraki, Technical University of Crete, Greece
Roland Ukor, FirstLinq Limited, UK
Torsten Ullrich, Fraunhofer Austria Research GmbH, Austria
José Valente de Oliveira, Universidade do Algarve, Portugal
Dieter Van Nuffel, University of Antwerp, Belgium
Shirshu Varma, Indian Institute of Information Technology, Allahabad, India
Konstantina Vassilopoulou, Harokopio University of Athens, Greece
Miroslav Velev, Aries Design Automation, USA
Tanja E. J. Vos, Universidad Politécnica de Valencia, Spain
Krzysztof Walczak, Poznan University of Economics, Poland
Jianwu Wang, San Diego Supercomputer Center / University of California, San Diego, USA
Yandong Wang, Wuhan University, China
Rainer Weinreich, Johannes Kepler University Linz, Austria
Stefan Wesarg, Fraunhofer IGD, Germany
Sebastian Wieczorek, SAP Research Center Darmstadt, Germany
Wojciech Wiza, Poznan University of Economics, Poland
Martin Wojtczyk, Technische Universität München, Germany
Hao Wu, School of Information Science and Engineering, Yunnan University, China
Mudasser F. Wyne, National University, USA
Zhengchuan Xu, Fudan University, P.R.China
Yiping Yao, National University of Defense Technology, Changsha, Hunan, China
Stoyan Yordanov Garbatov, Instituto de Engenharia de Sistemas e Computadores - Investigação e Desenvolvimento, INESC-ID, Portugal
Weihai Yu, University of Tromsø, Norway
Wenbing Zhao, Cleveland State University, USA
Hong Zhu, Oxford Brookes University, UK
Qiang Zhu, The University of Michigan - Dearborn, USA

International Journal on Advances in Software
Volume 7, Numbers 3 & 4, 2014

CONTENTS

Framework for Adaptive Sequential Pattern Recognition Applied on Credit Card Fraud Detection in the Online Games Industry

Michael Schaidnagel, Thomas Connolly
School of Computing
University of the West of Scotland
B00260359@studentmail.uws.ac.uk
Thomas.Connolly@uws.ac.uk

Fritz Laux
Data Management Lab
Reutlingen University
Fritz.Laux@reutlingen-university.de

***Abstract*—Online credit card fraud presents a significant challenge in the field of eCommerce. In 2012 alone, the total loss due to credit card fraud in the US amounted to $ 54 billion. Especially online games merchants have difficulties applying standard fraud detection algorithms to achieve timely and accurate detection. This paper describes the special constrains of this domain and highlights the reasons why conventional algorithms are not quite effective to deal with this problem. Our suggested solution for the problem originates from the fields of feature construction joined with the field of temporal sequence data mining. We present feature construction techniques, which are able to create discriminative features based on a sequence of transaction and are able to incorporate the time into the classification process. In addition to that, a framework is presented that allows for an automated and adaptive change of features in case the underlying pattern is changing.**

Keywords-feature construction, temporal data mining, binary classification, credit card fraud

I. INTRODUCTION

This work is an extension of our work in the field of fraud detection [1]. The approximate global business volume of the computer gaming industry in total rose from $ 79 billion in 2012 to $ 93 billion in 2013 [2]. It is estimated to reach $ 111 billion in 2015. New technology developments, such as browser games and Massive Multiplayer Online Games have created new business models (based on micropayments) for online games merchants. Both, technology and business model affect the customers payment behaviour. Their first choice for performing online payments is the credit card. The downside of this development is an increase in online credit card fraud, which continues to pose a big threat for online merchants. Over all branches, the total loss due to credit card fraud rose to $ 54 billion in 2013 in the US alone [3] and is supposed to increase further. Especially merchants in the online games industry are having difficulties applying standard techniques for fraud detection. The reason for this is the lack of personal information about their customers as well as the need for real time classification. Other online retailers (e.g., for fashion, books) are able to compare the shipment address with the billing address of an order to assess if a suspicious transaction is placed. In addition to that, online retailers have more time for extensive risk checks and can also have the manpower to verify customer´s identify by phone for high amount orders.

Online game merchants dealing with low volumes micro transactions and need to deliver instantly in real time. Therefore, an automated data mining approach needs to be considered in order to deal with fraudulent credit card transactions. The advantage of games merchants is their data situation. Data about the in-game behaviour as well as the earlier transactions can be collected along the way. Therefore, it is obvious to use the so far collected data to determine genuine and fraudulent behaviour patterns.

The collected transaction sequences have a more complex structure that other tuple based data. It is collected over time and incorporates timestamps of the particular items bought. We apply feature construction techniques to incorporate the temporal dimension of previous transactions into the classification process and therefore aid finding distinctive sequential patterns. A sequential pattern is only visible in the course of time [4].

The rest of the paper is structured as follows: Section II gives a short introduction in to the field of temporal sequence data mining to set the frame for the given set of problems. This is followed by Section III, which explains feature construction and feature selection. These research fields are later used to retrieve information from data sequences. The consecutive Section IV will define the problem at hand, introduces problem-related terms and describes the contribution of this work. Section V will then give an overview of the related work, which describes different data mining algorithms that are normally suggested for the given problem set. The algorithms are also part of the experimental evaluation. Section V will detail our suggested method and describe the major components. The suggested method is applied to a real life data set in Section VII to show its classification abilities. Section VIII concludes the results and mentions a few points for future development.

II. FROM DATA MINING TO TEMPORAL DATA MINING

Data Mining is a multidisciplinary research field, which uses methods and knowledge from many areas, such as database technology, machine learning, information theory, statistics, data visualization, artificial intelligence, and computing [5]. This created a *"solid science, with a firm mathematical base, and with very powerful tools"* [6] p. 5. It is the natural result of the evolution of information technology. This development started in the 1980´s when data access techniques began to merge, the relational data model was applied, and suitable programming languages

were developed. The following decade included the next significant step in data management: the use of Data Warehouses and Decision Support Systems. They allowed manipulation of data originating from several different sources and supported a dynamic and summarizing data analysis. However, these systems are not able to find more hidden patterns in data-rich but information-poor situations. This requires advanced data analysis tools which are provided by the research field of data mining [7]. In contrary to Data Warehouses and Decision Support Systems, Data Mining is computer-driven. It solves the query formulation problem. This means discovering patterns, which a user is not able to put into a database query or is only able to formulate the problem [7].

The term Temporal Data Mining refers to an emerging research issue [8], that is defined by Lin, Orgun, and Williams [9] p. 83 as *"a single step in the process of Knowledge Discovery in Temporal Databases that enumerates structures (temporal patterns or models) over the temporal data, and any algorithm that enumerates temporal patterns from, or fits models to, temporal data […]"*. This step increased its importance, since recent advances in data storage enabled companies to keep vast amounts of data that is related to time [9]. Temporal Data Mining is concerned with inferring knowledge from such data. Thereby the following two inference techniques can be applied [9]:

- Temporal deduction: inferring information that is a logical consequence of the information in the temporal database
- Temporal induction: inferring temporal information generalized from the temporal database

According to Kriegel et al. [10], Temporal Data Mining algorithms that are able to find correlations/patterns over time will play a key role in the process of understanding relationships and behaviour of complex objects. Complex objects in this case can be, for example, data sequences that contain several columns. A prime example therefore is transactions in an online shop. All transactions from one customer form a transaction sequence. Each transaction can contain information about the purchase date, the purchased items and the customers address etc. This work will show how behaviour over time can be captured in meaningful features and then be used for real time classification.

III. Feature Selection and feature construction

Today´s data sets can have billions of tuples and thousands of (often useless) attributes [11]. This development is often referred in the literature as the curse of dimensionality. The performance of classification algorithms can deteriorate if the wrong input is given and also the computational costs can increase significantly. The reason for this is the tendency of classifiers to overfit, if provided with misleading information. So in terms of the online shop example from above, the length of the street name of the customers address does not have an effect on the purchase behaviour and needs to be ruled out. In order to do this, feature selection techniques are applied on the given data to decrease data dimensionality. This results in an increase of classification performance and also in a reduction of the execution time.

Feature selection is defined as the process of selecting a subset of attributes from the original dataset, which allows a classifier to perform at least as good as with all attributes as input [11]. There are several different feature selection techniques that can be categorized as supervised, unsupervised and semi-supervised. Supervised techniques utilize the given label while unsupervised do not. In terms of feature selection strategies, there can be a categorization into filter, wrapper or hybrid models [12]. Filter models use certain criteria to assess features and select the features with the highest score. Wrapper models use clustering algorithms to find feature subsets and then evaluate these in terms of their clustering quality. The process is repeated until a suitable quality is reached. The heuristic search strategy is quite computationally expensive compared to the filter model. Hybrid models include a filtering step before the typical wrapper process in order to increase computational efficiency. Our presented work is using a filter model based on two measurements, which are further explained in Section V.B.

Feature construction on the other hand is defined as the process of discovering missing information about the patterns in data by inferring or creating additional attributes in order to aid the mining process [5], [7]. An example for a simple feature construction technique on a two dimensional problem could be the following: assume that A_1 is the width and A_2 is the length of a square. This can be transformed into a one-dimensional problem by creating the feature F as area $F = A_1 * A_2$ [13]. However, the success of feature construction is dependent on the target hypothesis of the Data Mining problem at hand. There is no use in calculating the area as a feature; if the pattern (that needs to be found) is connected to the aspect ratio of the squares. Shafti and Pérez [14] distinguish between two types of features construction techniques in terms of their construction strategy:

- hypothesis-driven: create features based on a hypothesis (which is expressed as a set of rules). These features are then added to the original data set and are used for the next iteration in which a new hypothesis will be tested. This process continues until a stopping requirement is satisfied.
- data-driven methods: create features based on predetermined functional expressions, which are applied on combinations of primitive attributes of a data set. These strategies are normally non-iterative and the new features are evaluated by directly assessing the data.

Our present work is using a data-driven construction strategy since it allows for an automated feature construction process that is not dependant on human intervention.

IV. Problem definition:

The problem of credit card fraud detection involves a number of constraints, which make it difficult to apply traditional algorithms.

Firstly, gamers do not feel comfortable to reveal their real life names and addresses in an online gaming environment. This lack of personal data, in addition to the short transaction histories of players, makes it difficult to apply standard Data Mining techniques.

Secondly, the real time nature of business makes it necessary to be able to apply an algorithm in real time, or near real-time, in order to reject fraudulent transactions at authorization time. Most of the techniques proposed so far are bulk oriented and designed for offline batch processing. Hence, it would be helpful to have a technique that can be integrated into already existing systems. This is also supported by Fayyad, Piatetsky-Shapiro, and P. Smyth [15] p. 49: *"A standalone discovery system might not be very useful"*.

Thirdly, cardinality of the occurrences in a dataset of the given domain is either too high or too low. On the one hand, there can be millions of different credit card numbers involved in the transactions, so that standard algorithms are not able to recognize the sequence structure of transactions of the same credit card. On the other hand, the in-game products sold in the online store can be very few, so that frequent pattern mining algorithms having a hard time if there are, for example, only 6 different products available.

Fourthly, the so far proposed methods are focusing on static vectors of attributes without any temporal evolution. Kriegel et al. [10] argues that due to historical reasons (i.e., given their static data during the 1980´s); many researchers created their algorithms only for static descriptions of objects and are therefore not designed to input data with dynamic behaviour. The inclusion of the dynamic properties of temporal data however, allows unveiling sequential patterns that occur in the course of time. An example for this in the field of credit card fraud detection is the current status of transaction. A transaction that has been approved by the credit card company (i.e., is booked as successful) can later be charged back. This change of status represents vital information that is revealed after some amount of time has passed.

The framework described in this article is able to overcome all the challenges described in this section. Key to success is the understanding of temporal sequence based patterns. However, the framework is currently focusing on the domain of credit card fraud detection, since it also contains domain-specific features (that will be described further down). The next section highlights our contribution to the body of knowledge. That is followed by a description of the characteristics of sequential data. The last subsection of this chapter will give a definition for feature interaction, which is a pattern that can be used by our proposed algorithm.

A. Contributions

There are several hints in the literature, which give suggestions on how to solve such a sequence classification problem that is presented by online credit card fraud: one direction of development in the field of temporal data mining is described by Lin, Orgun, and Williams [9]. They postulate a temporal sequence measure method that allows *"[…] an arbitrary interval between temporal points […]"* to create *"[…] a very powerful temporal sequence transformation method."* [9] p. 83.

Another direction of development is given by Tsai, Chen, and Chien [16]. According to their article, a sequential pattern classification problem can also be treated as a feature mining problem. This would allow feature mining algorithms to treat extracted patterns as features. However, Yang, Cao, and Liu [8] state that the well-known standard classification algorithms are difficult to be applied on sequential data due to vast number of potential features that can be generated out of a sequence. A solution for this could be to work on a different abstraction layer, i.e., to use a form of aggregation to simplify the data sequence into a row-based vector. This would then allow standard classification algorithms to be applied on complex sequential data.

Simplification is also suggested by Kriegel et al. [10] p. 90: *"Representing complex objects by means of simple objects like numerical feature vectors could be understood as a way to incorporate domain knowledge into the data mining process"*. However, he focuses more on the incorporation of domain expert knowledge into data mining. He postulates a technique to help domain experts *"to use the important features of an object to e.g., classify new objects of the same type, eventually by employing sophisticated functions to transform attributes of some type to features of some other type"* [10] p. 90 and to generalize this domain knowledge to keep pace with more complex ways of mining complex objects.

Our contribution to that research field is a novel algorithm that incorporates these ideas and puts them into one approach. The described feature construction techniques are able to include the time dimension during the aggregation of sequences. This allows using arbitrary time intervals as suggested by Lin, Orgun and Williams [9]. In a later step, the found features are normalized and arranged in a way that enables threshold based classification.

The framework presented in this work is able to handle the difficult data situation in the area of online credit card fraud detection. In addition to that, the framework is able to adapt to changes of the fraudulent behaviour. All necessary steps are described, starting from data preparation to feature construction to applying the right threshold.

In order to assess transactions without any history, a concept of cultural clusters is introduced to help classifying those transactions. In addition to that, a metric for assessing the suitability of features as well as the calculation of the threshold are introduced. The suggested approach was pitched against other algorithms on a real life data set. It was able to perform 16.26 % better than the best standard method (Bayesian Net) and achieves an almost perfect 99.59 % precision.

B. Characteristics of sequential data

Sequential Data is defined in the literature as a series of nominal symbols from a defined alphabet. This list of objects is normally registered within a domain ontology according to Adda et al. [17] as well as Antunes and Oliveira [18]. The term sequence must not be confused with the term time

TABLE I. SCHEMA OF SEQUENTIAL DATA

r	t	s_{id}	a_1	a_2	...	a_i	s_{label}
r_1	t_1	s_{id_1}	a_{1_1}	a_{2_1}	...	a_{i_1}	0
r_2	t_2	s_{id_1}	a_{1_2}	a_{2_2}	...	a_{i_2}	0
r_3	t_3	s_{id_1}	a_{1_3}	a_{2_3}	...	a_{i_3}	0
r_4	t_4	s_{id_2}	a_{1_4}	a_{2_4}	...	a_{i_4}	1
r_5	t_5	s_{id_2}	a_{1_6}	a_{2_5}	...	a_{i_5}	1
...	...	...	...	...	...	...	...
r_m	t_m	s_{id_n}	a_{1_m}	a_{2_m}	...	a_{i_m}	...

series, which is a sequence of continuous, real-valued elements. The research work proposed in this article is focusing on transactional datasets, which include information about the time of the transaction and can be attributed to logical units (i.e., sequences). This logical unit in the field of credit card fraud detection is the customer id. Every action can be represented in a data base as a row r, which has several attributes (i.e., columns). Each row is provided with a timestamp *t*. The attributes $a_i \in E$ of a row can be associated to a logical unit s_{id} (i.e., the customer_id). There are n sequences s_{id_n} in a data set E. Each sequence s_{id_n} consists of at least one row r. The number of rows in a sequence equals the length of a sequence ls, so that $1 \leq ls \leq m$.

Table I depicts the general schema of sequential data: It is important to differentiate between the number of rows (or tuples) m of a data set and the number of sequences n. Sequence s_1, from the example below, has a length ls = 3 and can be described as matrix such as $s_1 = \begin{pmatrix} a_{1_1} a_{2_1} \ldots a_{i_1} \\ a_{1_2} a_{2_2} \ldots a_{i_2} \\ a_{1_3} a_{2_3} \ldots a_{i_3} \end{pmatrix}$

C. Feature Interaction

The proposed framework is able to use interrelations among attributes of a dataset: It is possible, that the original data is not sufficient to adequately describe such an eventually existing interaction among attributes. Thereby interaction means that *"the relation between one attribute and the target concept depends on another attribute"* [19] p. 246. If the existing dependency is not constant, the interaction is called complex. An example of a complex interaction between two attributes in an instance space is shown below in Figure 1.

The '+' and '-' signs depict the distribution of the class labels of instances. So in the case of the left hand side, instances with a high vale for a_1 and a_2 have the label '-'. Interactions among data in general pose a problem for classifiers, since neither a_1 nor a_2 by itself contains enough information to distinguish between the labels.

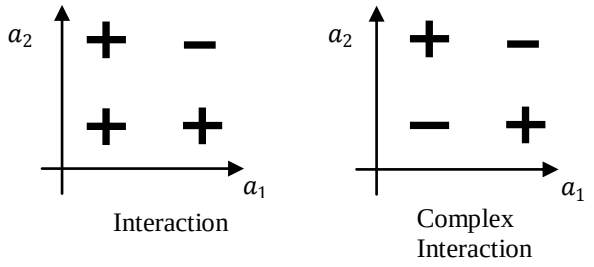


Figure 1. Schematic representation of complex feature interaction, based on Shafti and Pérez [19] p. 246

V. PROPOSED FRAMEWORK

The proposed framework is designed for classification tasks on data sequences consisting of transactions. Originally, the used features were constructed manually and incorporated domain specific knowledge [1]. The used feature construction techniques were automated, enriched, and generalized in a later step, also shown in Schaidnagel and Laux [42]. These techniques (also briefly discussed in Section V.A) are now combined to create an adaptive algorithm that is able to attune to changing fraud behaviour. Figure 2 shows an overview. The framework consists of two systems: the first one processes the credit card transactions and executes the classification (i.e., the fraud / non-fraud decision). The decision is based on a signal value that is calculated using the transaction history of the corresponding user account. The calculation is carried out by the feature assembler, which uses a formula (described in Subsection V.D) that consists of a multitude of features. The features are templates for how to aggregate a given sequence. They are provided from the feature pool, which is kept up to date by the second system.

The second system hosts the feature construction algorithm, which is briefly described in Subsection V.A. After a certain period of time, e.g., a week or a month, the second system uses a sliding window to query for training data, which is used to create new features. They are assessed using feature selection (also described in Subsection V.B). The performance of the newly constructed features is compared to the older ones in the feature pool and replaced if necessary.

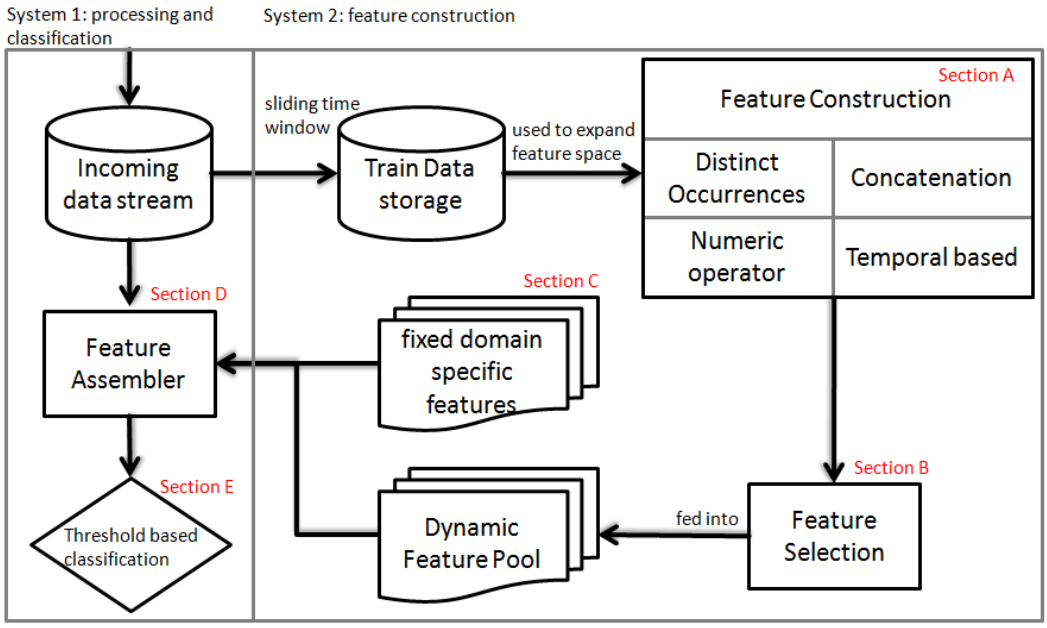


Figure 2. Framework Overview, Section A-E refer to the subsections in the description of Section VI.

A. *Feature Construction for transactional data*

As briefly described above, training data is periodically drawn from the execution environment, which is then used for 'training'. The training process consists of constructing and selecting suitable features that are able to distinguish between the two given labels. The columns of the original data set are called attributes, while the constructed data is called features.

The feature construction techniques that we used for this work utilizes a data-driven approach. It is in detail described in [42]. The data set needs to be annotated in an initial step. This is done by selecting an attribute s_{id} of the original data set that is used as a sequence identifier column for sequence aggregation. It identifies events/objects that can be logically associated to one entity. An example for such an attribute could be the account number or email address of a user. For the domain of credit card fraud detection, we use the term sequence to refer to all transactions belonging to a certain user email address. Please note that in the feature construction step, the transactions of a sequence are sorted by the transactions timestamp prior to aggregation. In a next step the user has to select two more columns: t and s_{label}. The timestamp column t is used to calculate the time elapsed between the collected data points of a sequence. The column s_{label} contains the binary target hypothesis. The label is sequence based, which means that every sequence must only have transaction of the same label value. In the domain of credit card fraud detection this means that all transactions of a user carry the genuine label until one transaction is charged back. Then the label of the sequence (i.e., all associated transactions) changes to fraudulent.

In a next step, we formulated feature construction techniques, which are able to create distinctive features if such a pattern is hidden in the data. We found four different feature construction techniques, which will be briefly described in the following subsections:

1) *Features based on distinct occurences*

A first approach for detecting sequential patterns is to investigate the number of distinct occurrences per sequence. It is possible that one target label has a higher variety in terms of occurrences than the other. This variety can be assessed by aggregating all sequences s_{id_n} of an attribute a_{i_n} and count the number of distinct occurrences, so no duplicates are counted. This can be applied on all string as well as numeric attributes of a data set. The results for all sequences s_{id_n} and attributes a_{i_n} are stored in an intermediate feature table and are assessed in terms of their suitability for classification in step B of the proposed framework.

2) *Concatenation based features*

A way to highlight interactions between two attributes is to concatenate them. Therefore, we systematically concatenate every string attribute in pairs of two and then again, count the distinct value-pairs per sequence identifier. Thereby interactions such as, if a_1 AND a_2 have low value pair variety for label 0, but a high value-pair variety for label 1, are highlighted. Even for data sets with a high number of different occurrences, this kind of feature construction will highlight distinct occurrences between both labels.

This technique can be applied on all string attributes of the given dataset. This simple technique is similar to most common column combinations that are described widely in the literature (e.g., [38], [40], [41]). However, we once again use this technique on a different abstraction layer since we aggregate by the sequence identifier s_{id}.

3) Numeric operator based features

Interactions among data can also occur between two numeric attributes. It is possible to capture such a pattern by combining two numeric attributes with basic arithmetic operators such as ”+”, ”-”, ”*” or ”/”. García [39] and Pagallo [37] describe such a technique for feature construction with fewer operators. Our approach incorporates more arithmetic operators and again, uses the sequence identifier attribute to aggregate the constructed features for each sequence. Let us put this into an example: attributes a_i and a_j are combined with the multiplication operator ”*” for a sequence s_{id_1}. The resulting feature $f = a_i * a_j$ is derived from the sequence

$$s_{id_1} = \begin{pmatrix} a_{i_1} & a_{j_1} \\ a_{i_2} & a_{j_2} \\ a_{i_3} & a_{j_3} \end{pmatrix} \quad (1)$$

To construct f we have to multiply each ‘row’ in the sequence and sum up the results: $f = (a_{i_1} * a_{j_1} + a_{i_2} * a_{j_2} + a_{i_3} * a_{j_3})$. If there is an interaction between two attributes for a certain target label, it will affect the resulting sum and can be measured (as described further in Section B.1). This process is repeated for all possible combinations of numeric attributes and for all of the above mentioned arithmetic operators.

4) Temporal based attributes

Patterns in sequences can also occur over time. Therefore, we created a feature construction technique that is able to use the time axis, which is incorporated in each sequence by the timestamp column t. This feature construction technique is applicable for both, numeric as well as string attributes. However, for string attributes, there need to be some preparations done, which are explained further down in this subsection. The process for numeric attributes basically multiplies the time interval (e.g., days, hours or minutes), between earliest data point and the current data point with the numeric value of the corresponding attribute, which results in a weighting. A hypothetical example is depicted in Table II.

The example shows two attributes a_i and a_j for two sequences ($s_{id} = 1$ and $s_{id} = 2$) as well as the t column. In order to calculate the temporal based feature f_p for attribute sequence $s_{id} = 1$ in terms of attribute a_i, we first have to calculate the time between the earliest data point of $s_{id} = 1$ and each of the ’current’ data points. This is depicted in Table II by the $\Delta time\ in\ days$ column. The next step is to multiply the value of each t_i in $s_{id} = 1$ with its corresponding delta time value: $(a_{i_1} * 1, a_{i_2} * 11, \dots, a_{i_4} * 24)$. The sum of this value is the new time based constructed feature f_p. This technique can be applied on all numeric attributes.

TABLE II. EXAMPLE FOR CREATING TEMPORAL BASED FEATURES

s_{id}	t	min(t) / s_{id}	$\Delta time$ *in days*	a_i	a_j	s_{label}
1	01.01.2013	01.01.2013	1	a_{i_1}	a_{j_1}	0
1	10.01.2013	01.01.2013	11	a_{i_2}	a_{j_2}	0
1	15.01.2013	01.01.2013	16	a_{i_3}	a_{j_3}	0
1	23.01.2013	01.01.2013	24	a_{i_4}	a_{j_4}	0
2	24.01.2013	01.01.2013	1	a_{i_5}	a_{j_5}	1
2	28.01.2013	01.01.2013	5	a_{i_6}	a_{j_6}	1
2	30.01.2013	01.01.2013	7	a_{i_7}	a_{j_7}	1

To use temporal based feature construction on string attributes, we need to incorporate an intermediate step. During this step we replace the string value by its posterior probability $p(\theta|x)$ (see also Hand [43], pp. 117-118 and pp. 354-356). The posterior probability is the probability of an occurrence a_i, given that its label $s_{label} = 1$ divided by the overall number of that occurrence $p(a_n)$. The probability is based on the distribution of the occurrences in the training data:

$$p(a_i|s_{label} = 1) = \frac{p(a_i|s_{label} = 1)}{p(a_n)} \quad (2)$$

It is possible that there is a pattern within the data that can be characterized by certain occurrences. This means that some occurrences have great tendency towards one of the target labels (i.e., having a high probability for one label). The above described technique allows us to make this pattern visible by multiplying the posterior probability with the temporal axis of the given sequences.

However, it is also possible that the number of distinct occurrences of a string attribute is too high. This will lead to very small posterior probabilities that make it difficult to create meaningful and distinctive features. In such cases, it is recommended to take the logarithm of the posterior probability for cases with high cardinality.

B. Feature Selection

The feature construction techniques described in previous section generate a vast amount of features, which need to be assessed if they are useful for classification. Therefore, the next step in our framework (see also Figure 2) is dealing with feature selection.

Feature selection in general is an important step in the KDD process. The performance of classification algorithms can deteriorate, if the wrong input is given and also the computational costs can increase tremendously. Reason for the deterioration in performance is the tendency of classifiers to overfit, if provided with misleading information. In order

to avoid this, data miners created methods such as feature selection to decrease dimensionality of the data and as a result of that, increase classification performance and also the execution times.

A supervised filter model (see also Charu and Chandan [12]) is adopted in our framework to find the most suitable features created. There are two measurements that we used for assessing whether a constructed feature is suitable to distinguish between the two given label. The assessment is executed by applying a user defined threshold for both measurements. It is favourable to start with high thresholds, since they allow only the most distinguishing features. This also keeps the feature space, which needs to be constructed during classification, on a manageable level. If the classification performance of the top features is not satisfactory, the threshold can be subsequently lowered.

1) Split

Goal for this feature selection measurement is to find features that are 'in general' suitable for distinction between the two given labels. The average of the features for both groups (i.e., the two given labels) is calculated. The average is a sort of centre for the two clusters. The so-called split value is calculated by measuring the normalized distance between the two cluster-centres as it can be seen in (5)

$$avg_0 = avg(\{f_p \in S | s_{label} = 0\}) \quad (3)$$

$$avg_1 = avg(\{f_p \in S | s_{label} = 1\}) \quad (4)$$

$$split_{f_i} = \frac{|avg_0 - avg_1|}{avg_0 + avg_1} \quad (5)$$

A large distance is thereby favoured. The advantage of calculating the average is that a few false positives within the data do not have such a big impact on the feature selection process. However, average calculation is prone to single extreme or erroneous values if the data is completely unprepared (data normalization would not help in that case).

2) Number of null values

The second feature selection measurement is the number of *NULL* values for each target label. This is a support measurement, which denotes if the achieved split value is based on many sequences or not. So there could be the situation that a constructed feature has a high split value, but might be useless since it cannot be used very often due to large number of *NULL* values for the particular features.

C. Fixed domain specific features

In order to maximize the amount of information retrieved from the transaction history, we incorporate domain specific features to the classification process. The concept of so called cultural clusters was introduced in order to help classifying transactions without any history. The basic idea is to get as much information out of the given attributes as possible. These attributes include the origin of the user (IP country) and the origin of the credit card used in a transaction (BIN country – BIN is an abbreviation for Bank Identification Number: The first 6 digits of a credit card number, enables to locate the card issuing bank of the cardholder). Countries are grouped together by an expert regarding their cultural proximity to each other. The clusters used in this work are roughly based on continents. A range of weights is assigned to each cluster. Every country is assigned with a specific weight within its cluster´s range depending on its cultural distance to its cluster centre and the risk of the county of being defrauded. The weight of a country within a certain cultural cluster is set empirically and can be subject for adaption, in case the fraudulent behaviour changes. In other words: the weight of a country lies within the range of its cultural cluster and is set by an initial value, based on the experience of a fraud expert. If cards from this country turn out to be defrauded frequently, the weight can be increased (within the limits of its cultural clusters). This will increase the risk value of a country pair, which can be calculated as it can be seen in (6):

$$risk = |weight\ (IP\ country) - weight(BIN\ country)| \quad (6)$$

This value will be low for country pairs within the own country cluster (e.g., a user from Sweden tries to use a card originated in Norway) or 0 if the user and the corresponding card are from the same country. On the other hand, this value increases if there is a suspicious country pair involved (e.g., cross-cultural cluster). This simple metric allows depicting complex risk relationships between several countries.

D. Feature Assembler

The previous subsections described how to construct and assess suitable features for the given fraud detection task. This subsection is about how to use these features to make use of feature interaction by assembling the respective features in a certain way. Prior to that, the features are normalized with the min-max normalization [5] to bring them on the same numerical level (ranging from 0 to 1). The Feature Assembler is part of System 1 and is invoked at the time a sequence of credit card transactions need to be classified. It uses the templates of features as input, which are currently held in the Dynamic Feature Pool as well as the fixed domain specific features (see also Figure 2). The templates of the features (i.e., the description on how to construct them), are then applied on the sequences in the incoming data stream that need to be classified. We thereby differentiate between two types of interactive features $f_i \in F$.

The first type of features $f_{i_{nom}}$ tends to 1 if normalized and will be summed up in the nominator of a fraction. The denominator, in contrary, is composed of the second type of features $f_{i_{denom}} \in F$, which tend to 0 if normalized with min-max normalization. If the quotient of the normalization expression is not defined, it will be discarded. The fraction depicted in (7), is used for calculating a signal value that can then be used for binary classification.

The assembling of the interactive features will result in a high signal value if the sequence in question is similar to the average of all sequences of the target label.

$$signal = \frac{\sum f_{i_{nom}}}{\sum f_{i_{denom}}} \quad (7)$$

E. Threshold selection and application

The resulting signal value from (7) is an indication on how predominant fraudulent behaviour is in the assessed sequence. As a last step of our framework, a threshold value, whose violation will lead to the classification fraudulent transaction, needs to be defined. This threshold is determined empirically by undertaking a series of experiments with a set of thresholds (e.g., from 0 to 100). We use accuracy metrics such as Precision P, Recall R and score F1 to assess each tested threshold.

Precision P is defined in the literature as (e.g., [44], [45]) the ratio of true positives (TP) and the total number of positives predicted by a model. That is in our case the number of genuine transactions that have been declared to be genuine plus the number of fraudulent transaction that also have been labelled genuine:

$$P = \frac{TP}{(TP+FP)} \quad (8)$$

Recall R on the other hand is defined as the number of true positives divided by the sum of true positives and false negatives. In our example, we have to divide the number of fraudulent transactions detected by our model by the sum of the detected fraudulent transactions (TP) and the not detected fraudulent transactions (FN):

$$P = \frac{TP}{(TP+FN)} \quad (9)$$

The measurement F1 represents the harmonic mean of Precision and Recall and is used to rank the performance of different methods in the experimental evaluation:

$$F1 = 2 * \frac{P*R}{P+R} \quad (10)$$

The development of these performance measures over different threshold values is depicted Figure 3 for an example case.

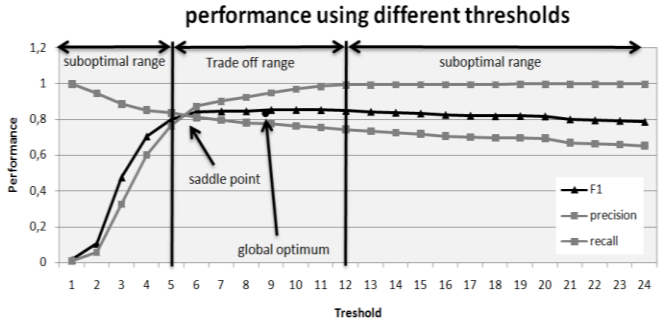


Figure 3. Determining threshold value

Accuracy indicators are increasing fast until threshold value 5. It is not reasonable to select a threshold lower than 5, since the F1 is far from the global optimum. From threshold 5 on, there is an intersection point, which will keep the F1 near the global optimum. This second range can be called "trade-off range" and spans up to threshold value 12, in the case depicted in Figure 3. Within this range the merchant can choose between detecting more fraudsters, including a higher rate of false positives or catching less fraudsters, but increase Precision and therefore avoid false positives. This choice can depend on the ability of the merchant to deal with false positives and on the merchants specific total fraud costs. In the context of fraud detection the term total fraud costs means the sum of lost value, scanning cost as well as reimbursement fees associated with a fraud case.

After a certain threshold value, in the shown case 12, the Precision is almost 1 and will only increase insignificantly. The Recall and consecutively F1, will decrease from that point. Reason for this is the intrinsic mechanic in the used formula. Fraudulent transactions with a comparable low fraud profile will be assigned a lower risk level. This level will hopefully be still higher than the risk level of genuine users. If however, the threshold is set high enough these lower profile fraud cases will be classified incorrectly as genuine. This will cause the Recall and F1 to drop. Hence, it makes no sense to choose a threshold greater than 12.

VI. Related Work

So far, there have been many standard data mining algorithms applied in the field of credit card fraud detection [5]. Please note that we do not go into details here on how they work. All mentioned methods have been implemented and will be compared in terms of fraud detection performance in Section VII.

Artificial Neural Network (ANN): Gosh and Reilly [20] were the first ones to adapt Neural Networks on credit card fraud detection. Other authors such as Dorronsoro et al. [21], Brause et al. [22] and Maes et al.[23] have also implemented ANNs in real life applications. ANNs in general are too dependent on meaningful attributes, which might not necessarily be available. The information gain from such attributes is too low to be utilized in ANNs.

Bayesian Belief Network (BBN): The first implementation for fraud detection was done by Ezawa et al. [24]. Other recent implementations are Lam et al. [23], Maes et al. [23] and Gadi et al. [25]. However, some data set do not provide enough attributes in order to construct a suitable network.

Hidden Markov Model (HMM): In recent years several research groups applied this model for fraud detection. Srivastava et al. [26] have conducted a very systematic and thorough research in their work. Other implementations were done by Mhamane et al. [27], Bhusari et al. [28] as well as Dhok and Bamnote [29]. A classic and comprehensive introduction to the topic of HMM was published by Rabbiner and Juang [30] and also Stamp [31] is worth reading for introductory purposes. HMMs in general are only able to utilize a single numeric attribute for their prediction, which is insufficient for a proper classification.

Decision Tree (DT): The biggest impact on how Decision Trees are built had Quinlan [32] in the late 90s. There have been some applications on fraud detection in recent years, e.g., Minegishi et al. [33]. Other mentionable fraud detection implementations are Sahin and Duman [34], Sherly et al. [35] and Gadi et al. [25]. DTs in general suffer the same insufficiencies as ANNs.

Support Vector Machine (SVM): Li and Sleep [49] use a Support Vector Machine for sequence classification. In essence, they compare similarity using a kernel matrix. Their

similarity measure is based on n-grams of varying length. The problem of exploding features generation complexity is alleviated by the use of LZ78 algorithm. The constructed features are not only simple binary presence/absence bits, but so-called relative frequency counts. This is assumed to create a finer grain of features. They also use a weighting scheme to highlight discriminative, but infrequent patterns. Dileep and Sekhar [48] describe an intermediate matching kernel for a SVM to help classification of sequential patterns.

Earlier work in the field of feature construction was done by Setiono and Liu [36]. They used a neuronal network to construct features in an automatic way for continuous and discrete data. Pagallo [37] proposed FRINGE, which builds a decision tree based on the primitive attributes to find suitable Boolean combinations of attributes near the fringe of the tree. The newly constructed features are then added to the initial attributes and the the process is repeated until no new features are created. Zupan and Bohanec [38] used a neuronal net for attribute selection and applied the resulting feature set on the well known C4.5 [32] induction algorithm. Feature construction can also be used in conjunction with linguistic fuzzy rule models. García et al. [39] use previously defined functions over the input variables in order to test if the resulting combination returns more information about the classification than the single variables.

However, in order to deal with the increasing complexity of their genetic algorithm in the empirical part, García et al. only used three functions ($SUM(x_i,x_j)$, $PRODUCT(x_i,x_j)$, $SUBSTRACT\_ABS(x_i,x_j)$) to enlarge the feature space. Another approach to feature construction, which utilizes a genetic algorithm, is described by Sia and Alfred [40]. Although, his approach is not using different functions to create new combinations of features, it can create a big variety of features since it is not limited to binary combination. The method is called FLFCWS (Fixed-Length feature construction with Substitution). It constructs a set that consist of randomly combined feature subsets. This allows initial features to be used more than once for feature construction. That means that it is able to combine more than two attributes at a time. The genetic algorithm selects thereby the crossover points for the feature sequences. Another mentionable contribution to the field of feature construction was done by Shafti and Pérez [41]. They describe MFE3/GA, a method that uses a global search strategy (i.e., finding the optimal solution) to reduce the original data dimension and find new non-algebraic representations of features. Her primary focus is to find interactions between the original features (such as the interaction of several cards in a poker game that form a certain hand).

Lesh, Zaki and Ogihara [46] present FeatureMine - a feature construction technique for sequential data. It combines two data mining paradigms: sequence mining and classification algorithms. They understand sequences as a series of events. Each event is described by a set of predicates, e.g. AB --> B --> CD. There is also a timestamp associated with each event. FeatureMine starts by mining frequent and strong patterns. Frequency is defined by a threshold that is specified by the user. Strong is defined as a confidence level that needs to be over a user specific threshold. The found sequences are pruned and selected using some heuristics. The prevailing sequences lattices are stored in a vertical database layout. The constructed features have been feed into the Winnow and Naive Bayes classification algorithms. However, this approach only creates frequent itemsets and is not applicable on transactional data.

Shafti and Pérez [47] present MFE3/GA, which is a feature construction technique that is able to detect and encapsulate feature interactions. The encapsulation is what allows classifiers to deal with interacting features. MFE3/GA in essence searches through the initial space of an attribute subsets to find subset of interaction attributes as well as a function over each of the found subsets. The suitable functions are then added as new features to the original data set. The C4.5 learner is then applied for the data mining process. So far only nominal attributes are being processed, so that class labels and binary/continuous attributes need to be normalized. A feature is in this context is a bit-string of length N, where each bit shows the presence or absence of one of the N original attributes. This form of representation reduces the complexity if elaborate features are constructed. The number of subsets within each feature is limited by a parameter, which is defined by the user. The bit representation of data is not sufficient to model real-world transactional data.

VII. Experimental Evaluation

The performance of the proposed feature construction techniques are compared to the standard techniques, mentioned in the related work. This comparison is based on real credit card fraud data, which was thankworthy provided by a successful gaming company in the online games market. Unfortunately, the given data did not span a long enough time frame to show the adaption capabilities of the presented feature assembler. Hence, Subsection VII.C shows a synthetic example for a pattern that is changing over the course of time.

A. Data Set

The given credit card fraud data set comprises of 156,883 credit card transactions from 63,933 unique users. The records in the data set have the schema as it can be seen in Table III. Due to the high number of occurrences in several

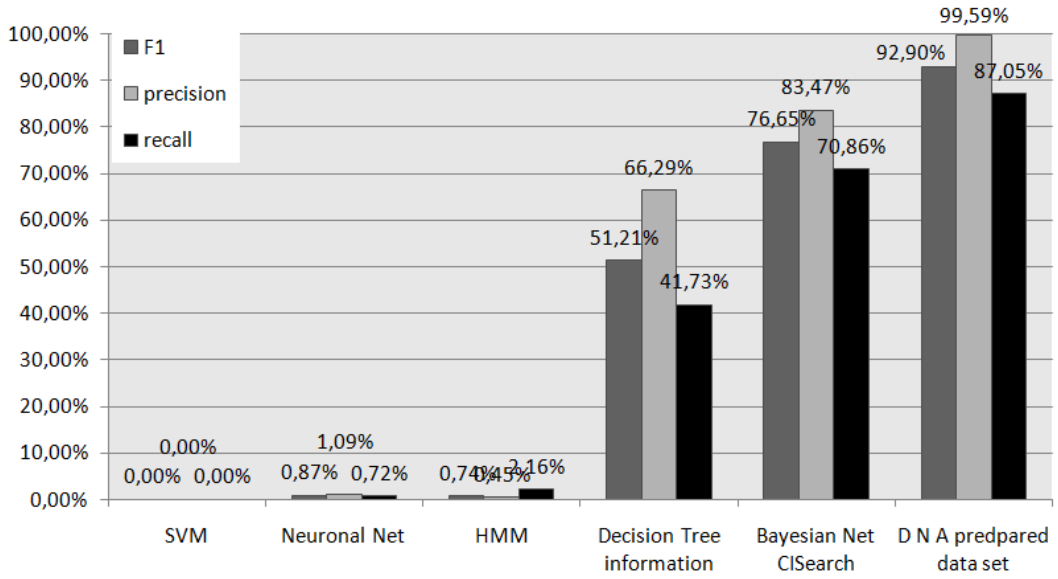


Figure 4. Fraud detection performance comparison

TABLE III. FULL DATA SET SCHEMA

column Name	description
created	timestamp of the payment transaction
user_signuptime	Time a user startet the game
creditcard_token	identifies credit card, hashed
card_bin	Bank Identification Number
user_country	user´s land of origin
user_id	User identification number
user_email	hashed for privacy compliance
transaction_amount	Volume of purchase
order_payment_status	Status of order

TABLE IV. ADDITIONAL FEATURES OF THE PREPARED DATA SET

column Name	description
bin_country	2 letter country code derived from card_bin
days_since_signup	integer attribute calculated as difference from signup_time to created
total_count	denotes total transaction figure for a particular user_email
package	a single letter attribute ranging from A to E. It was derived from the offer_price attribute to reduce the cardinality of the offer_price attribute

columns as well as the lack of distinctive attributes, most of the standard algorithms were not applicable on that data set right away. In order to get a fair comparison and to overcome these obstacles, several adaptations to the data set were made. The resulting prepared data has a minimum sequence length of three (smaller sequences have been discarded) and four derived attributes (see Table IV) were added.

The prepared data set comprised of 13,298 unique users, which are accompanied by 46,516 transactions. The data label distribution in the data set is heavily skewed, which means that there are ~99 % of genuine transactions. The last transaction of each user was cut out in order to form the test data set. This procedure segmented the prepared data set into 71.4 % train data and 28.58 % test data.

B. *Fraud Detection Performance*

All tests in this section were performed using the prepared data set. We used the F1 score in order to rank the compared methods. As shown in Figure 4, the proposed approach is able to perform 16.25 % better than the best standard method, which is the Bayesian Net. The SVM was not able to detect the pattern within the rather short sequences. Hence, it defaulted and predicted genuine for all transaction. This resulted in a F1 measure of 0.00 %. Main reason for the poor performance of HMM´s in credit card fraud detection for online gaming merchants is the very low sequence length. These models are successful at credit card issuing banks since their sequence length enfolds the entire history of the cardholder. The HMM is also not able to properly use the time elapsed during the transactions. The neural nets were performing poor due to their focus on just the tuple level of the underlying data. They were not able to incorporate the sequence dimension into the model.

There have also been additional experiments with various combinations of neurons and different learning rates were carried out. All experiments resulted in the weak performance as shown above. The algorithm proposed in this article is also able to achieve an almost perfect 99.59 % Precision, which is especially valuable for online gaming merchants, since it reduces the risk of punishing genuine users and consecutively reduces the risk of reputation loss.

C. Adaptive Framework example

This subsection describes a small example that will show the adaptive capabilities of the proposed framework. To keep things simple, we left out the sequential part of the construction algorithm and just focus on a two dimensional problem (rectangle classification).

Assume that we are given data about rectangles as shown in Table V. Given are the two attributes width and length of the rectangles as well as the label column. Goal is to be able to distinguish between the two given classes. We can then use this data to create several features, as described in Section A. Please note that for the sake of simplicity in this case we only create some operator numeric based features. The constructed features are depicted in Table VI. The constructed features are assessed by calculating the split value, as described in Section B and shown in Table VII. The highest split values have features f_3 and f_4. Reason for this is that these features are capturing the underlying pattern behind the labels: the blue rectangles are 'laying' (length <width) while the orange rectangles are 'standing' (length > width).

So we can now use features f_3 and f_4 for classification of new rectangles where the label is unknown (e.g., we set up a threshold of 1 for feature f_4 so that $f_4 \geq 1 \rightarrow blue$). The blueprint of f_4 and the threshold are forwarded to the feature assembler, which is used to execute the classification.

Now let us assume that some time has passed and new training data is handed into the features construction system. Table VIII depicts the new data. Again, we apply the feature construction algorithm on the data (depicted in Table IX) and calculate the split value (depicted in Table X). It can be seen that features f_3 and f_4 are not suitable anymore to distinguish between the two given labels. Reason for this is that the underlying pattern has changed. It seems like that the area of the rectangles is now useful for classification. The proposed framework can adapt to this change by sending the blueprints for feature f_1 to the feature assembler, which will then use the new selected features for classification.

TABLE V. RECTANGLE EXAMPLE DATA

id	width a_1	length a_2	label
#1	2	3	orange
#2	4	1	blue
#3	2	2	blue
#4	3	2	blue
#5	1	3	orange

TABLE VI. CONSTRUCTED FEATURES

id	width a_1	length a_2	label	$f_1 = a_1 * a_2$	$f_2 = a_1+a_2$	$f_3 = a_1-a_2$	$f_4= a_1/a_2$
#1	2	3	orange	6	5	-1	0.67
#2	4	1	blue	4	5	3	4
#3	2	2	blue	4	4	0	1
#4	3	2	blue	6	5	1	1.5
#5	1	3	orange	3	4	-2	0.33

TABLE VII. CALCULATING SPLIT VALUE

average	$a_1 * a_2$	a_1+a_2	a_1-a_2	a_1/a_2
blue	4.67	4.67	1.33	2.17
orange	4.5	4.5	-1.5	0.5
split	0.17	0.17	2.83	1.67

TABLE VIII. NEW TRAINING DATA

id	width a_1	length a_2	label
#6	2	1	orange
#7	3	2	orange
#8	5	3	blue
#9	6	2	blue

TABLE IX. NEW CONSTRUCTED FEATURES

id	width a_1	length a_2	label	$f_1 = a_1 * a_2$	$f_2 = a_1+a_2$	$f_3 = a_1-a_2$	$f_4 =a_1/a_2$
#6	2	1	orange	2	3	1	2
#7	3	2	orange	6	5	1	1.5
#8	5	3	blue	15	8	2	1.67
#9	6	2	blue	12	8	4	3

TABLE X. NEW FEATURE SELECTION

average	$a_1 * a_2$	a_1+a_2	a_1-a_2	a_1/a_2
blue	13.5	8	3	2.33
orange	4	4	1	1.75
split	9.5	4	2	0.58

VIII. CONCLUSION

Data pre-processing and selection are very important steps in the data mining process. This can be challenging, if there is no domain expert knowledge available. The framework proposed in this work aims to give guidance on how to systematically find knowledge in data by using an automated feature construction algorithm. In addition to that it shows how these features can be used for binary classification. The proposed automated feature construction algorithm is able to systematically find and assess suitable sequence based features for binary classification tasks. It thereby is able to utilize the time dimension in a sequence of actions in order to access information, which can have a significant impact on the discriminatory power of features. The feature assembling formula is an efficient way to store discovered patterns and use them without starting each time from scratch when a new transaction is added to the sequence.

The framework was applied on the problem of credit card fraud detection in online games. The problem is caused by the lack of useful financial data, the anonymity in online games as well as the comparably short transaction sequences. In addition, a domain specific concept of country clusters is used to evaluate the legitimacy of a transaction. The

proposed techniques were able to perform 16.25 % better than the best standard method (Bayesian Net) and achieve 99.59 % Precision. The achieved Recall rate (87.05 %) reduced the probability for false negatives and therefore the need for human intervention is reduced.

Future Work: The next steps in the development of the proposed algorithms and its associated techniques, is to apply it on other domains with similar specifications. Intrusion detection in networks or detecting DDOS attacks are both fields in which few attributes are available and behaviour over time is important.

The further development of the feature construction algorithm comprises of the implementation of further mathematical functions into the construction process. So it is possible to generate features with logarithm or exponential powers. It would also be possible to create features based on more than two attributes.

In terms of feature alignment, it would also be helpful to incorporate the sequence length into the algorithm. The algorithm may be susceptible to the sequence length due to the proposed additive technique depicted in Formula (7). The used data set did not allow us to precisely quantify possible impacts.

REFERENCES

[1] M. Schaidnagel and F. Laux, "DNA: an online algorithm for credit card fraud detection for games merchants," in *The Second International Conference on Data Analytics*, pp.1-6 (2014, Jan. 09).

[2] R. van der Meulen, Gartner Says Worldwide Video Game Market to Total $93 Billion in 2013. Available: http://www.gartner.com/newsroom/id/2614915 (2013, Feb. 03).

[3] F. Briggs, Fraud costs US retailers $54bn a year, according to new KountVolumatic 2013 survey. Available: http://www.kount.com/_blog/Press_Coverage/post/fraud-costs-usretailers-54-year-according/ (2013, Feb. 03).

[4] Y.-H. Hu, F. Wu, and C.-I. Yang, "Mining multi-level time-interval sequential patterns in sequence databases," in *Software Engineering and Data Mining (SEDM)*, 2010 2nd International Conference on, pp. 416–421.

[5] J. Han, M. Kamber, and J. Pei, *Data mining: Concepts and techniques*, third edition, 3rd ed. Waltham, Mass: Morgan Kaufmann Publishers, 2012.

[6] I. H. Witten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, Second Edition. Permissions Oxford, UK: permissions: Elsevier Inc, 2005.

[7] A. Symeonidis and P. Mitkaas, Agent Intelligence Through Data Mining: Data Mining and Knowledge Discovery: A Brief Overview: Springer US, 2005.

[8] Y. Yang, L. Cao, and L. Liu, "Time-sensitive feature mining for temporal sequence classification," in *11th Pacific Rim International Conference on Artificial Intelligence*, pp. 315–326.

[9] W. Lin, M. A. Orgun, and G. J. Williams, "An overview of temporal data mining," in *Proceedings of the 1st Australian data mining workshop (ADM02)*. Canberra, Australia, vol. 2002, pp. 83–90.

[10] H.-P. Kriegel, K. M. Borgwardt, P. Kröger, A. Pryakhin, M. Schubert, and A. Zimek, "Future trends in data mining," in *Data Min Knowl Disc*, vol. 15, no. 1, pp. 87–97.

[11] K.J. Cios, R.W. Swiniarski, W. Pedrycz, and L.A. Kurgan, Eds, The Knowledge Discovery Process: *A Knowledge Discovery Approach*. US: Springer US, 2007.

[12] A. Charu and R. Chandan, Eds, *Data Clustering: Algorithms and Applications: Feature Selection for Clustering: A Review*. CRC: Chapman and Hall, 2012.

[13] H. Liu and H. Motoda, *Feature Extraction, Construction and Selection: A Data Mining Perspective*, Dordrecht: Kluwer Academic Publishers, 1998.

[14] L. S. Shafti and E. Pérez, "Genetic approach to constructive induction based on non-algebraic feature representation," in *5th International Symposium on Intelligent Data Analysis*, IDA 2003, Berlin, Germany, August 28-30, 2003, pp. 599–610.

[15] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "From data mining to knowledge discovery in databases," *AI Magazine*, pp. 36–54.

[16] C.-Y. Tsai, C.-J.Chen, and C.-J. Chien, "A time-interval sequence classification method," *Knowl. Inf. Syst.*, vol. 37, no. 2, pp. 251–278.

[17] M. Adda, P. Valtchev, R. Missaoui, and C. Djeraba, "On the discovery of semantically enhanced sequential patterns," in *Machine Learning and Applications*, 2005, pp. 383–390.

[18] C. Antunes and M. L. Oliveira, "Temporal data mining: an overview," in *Workshop on Artificial Intelligence for Financial Time Series Analysis*, pp. 1–13.

[19] L. S. Shafti and E. Pérez, "Machine learning by multi-feature extraction using genetic algorithms," in *Advances in Artificial Intelligence – IBERAMIA 2004*, pp. 246–255.

[20] S. Ghosh and D. Reilly, Eds, *Credit card fraud detection with a neural-network*, 1994.

[21] J. Dorronsoro, F. Ginel, C. Sgnchez, and C. Cruz, "Neural fraud detection in credit card operations," in *IEEE Trans. Neural Netw*, Vol. 8, No. 4, pp. 827–834.

[22] R. Brause, T. Langsdorf, and M. Hepp, "Neural data mining for credit card fraud detection," in *Tools with Artificial Intelligence*, 1999, pp. 103–106.

[23] S. Maes, K. Tuyls, B. Vanschoenwinkel, and B. Manderick, "Credit card fraud detection using bayesian and neural networks," in *Proceedings of the 1st international asian congress on neuro fuzzy technologies*, 2002.

[24] K. Ezawa and S. Norton, "Constructing bayesian networks to predict uncollectible telecommunications accounts," in *IEEE Expert*, vol. 11, no. 5, pp. 45–51.

[25] M. F. A. Gadi, X. Wang, and A. P. d. Lago, "Comparison with parametric optimization in credit card fraud detection," in *Machine Learning and Applications*, 2008. ICMLA '08. Seventh International Conference on, pp. 279–285.

[26] A. Srivastava, A. Kundu, S. Sural, and A. Majumdar, "Credit card fraud detection using hidden markov model," in *IEEE Trans. Dependable and Secure Comput*, Vol. 5, No.1, pp. 37–48, 2008 (2013, Nov. 21).

[27] S. Mhamane and L. Lobo, "Fraud detection in online banking using HMM," in *International Proceedings of Computer Science & Information Technology*, 2012, *International Proceedings of Computer Science and Information Technology*, Vol. 37, pp. 200–204.

[28] V. Bhusari and S. Patil, "Study of hidden markov model in credit card fraudulent detection," in *IJCA*, Vol. 20, No. 5, pp. 33–36.

[29] S. S. Dhok and G. R. Bamnote, "Credit card fraud detection using hidden markov model," in *International Journal of Advanced Research in Computer Science*, Vol. 3, No. 3, 2012.

[30] L. Rabiner and B. Juang, "An introduction to hidden markov models," in *IEEE ASSP Mag*, Vol. 3, No. 1, pp. 4–16, 1986 (2013, Nov. 26).

[31] M. Stamp, "A revealing introduction to hidden Markov models," Department of Computer Science San Jose State University.

[32] J. R. Quinlan, *C4. 5: programs for machine learning*, Morgan Kaufmann Publishers, 1993.

[33] T. Minegishi and A. Niimi, "Detection of fraud use of credit card by extended VFDT," in *Internet Security (WorldCIS)*, World Congress on, 2011, pp. 152–159.

[34] Y. Sahin and E. Duman, "Detecting credit card fraud by decision trees and support vector machines," in *Proceedings of the International Multi Conference of Engineers and Computer Scientists*, 2011.

[35] K. K. Sherly and R. Nedunchezhian, "BOAT adaptive credit card fraud detection system," in *Computational Intelligence and Computing Research (ICCIC)*, 2010 IEEE International Conference on, pp. 1–7.

[36] R. Setiono and Huan Liu, "Fragmentation problem and automated feature construction," in *Tools with Artificial Intelligence*, Proceedings. Tenth IEEE International Conference on, 1998, pp. 208–215.

[37] G. Pagallo, "Learning DNF by Decision Trees," in *IJCAI*, 1989, pp. 639–644.

[38] B. Zupan, M. Bohanec, J. Demsar, and I. Bratko, "Feature transformation by function decomposition," in *IEEE Intell. Syst*, Vol. 13, No. 2, pp. 38–43.

[39] D. Garcia, A. Gonzalez, and R. Perez, "A two-step approach of feature construction for a genetic learning algorithm," in *Fuzzy Systems (FUZZ)*, 2011 IEEE International Conference on, pp. 1255–1262.

[40] F. Sia and R. Alfred, "Evolutionary-based feature construction with substitution for data summarization using DARA," in *4th Conference on Data Mining and Optimization* (DMO), 2012, pp. 53–58.

[41] L. S. Shafti and E. Pérez, "Data reduction by genetic algorithms and non-algebraic feature construction: A Case Study," in *Eighth International Conference on Hybrid Intelligent Systems*, pp. 573–578.

[42] M. Schaidnagel and F. Laux, "Feature construction for time ordered data sequences," in *Proceedings of the Sixth International Conference on Advances in Databases, Knowledge, and Data Applications*, 2014, Jan. 10.

[43] D. J. Hand, H. Mannila, and P. Smyth, *Principles of data mining (adaptive computation and machine learning)*, The MIT Press.

[44] Claude Sammut and Geoffrey I. Webb, *Encyclopedia of Machine Learning*: Precision and Recall. US: Springer, 2010.

[45] C. Goutte and E. Gaussier, "A probabilistic interpretation of precision, recall and f-Score, with Implication for Evaluation," in *27th European Conference on IR Research*, ECIR 2005, Santiago de Compostela, Spain, March 21-23, 2005. Proceedings, pp. 345–359.

[46] N. Lesh, MJ. Zaki and M. Ogihara, "Scalable feature mining for sequential data," in *IEEE Intelligent Systems*, Vol 2, 2000, pp. 48-56.

[47] L. S. Shafti and E. Pérez, "Feature construction and feature selection in presence of attribute interaction," in *4th International Conference HAIS 2009*, Salamanca, Spain, June 10-12, 2009, pp. 589-596.

[48] A. D. Dileep and C. Chandra Sekhar, "HMM based intermediate matching kernel for classification of sequential patterns of speech using support vector machines," in *IEEE Transactions on Audio, Speech, and Language Processing*, Vol. 21, No. 12, December 2013, pp. 2570-2582.

[49] M. Li and R. Sleep, "A robust approach to sequence classification," in *Proceedings of the 17th IEEE International Conference on Tools with Artificial Intelligence*, 2005, pp. 5-8.

Cloud-based Collaborative Software Development: A Concept for Managing Transparency and Privacy based on Datasteads

Roy Oberhauser
Computer Science Dept.
Aalen University
Aalen, Germany
roy.oberhauser@htw-aalen.de

Abstract - **Cloud-centric collaboration in (global) software development continues to gain traction, resulting in new development paradigms such as Tools-as-a-Service (TaaS) and Cloud Development Environments based on Software-as-a-Service (SaaS). However, both within and between clouds, there are associated security and privacy issues to both individuals and organizations that can potentially hamper such well-intentioned collaboration. This paper describes an inter-cloud security and privacy concept for heterogeneous cloud developer collaboration environments that pragmatically addresses the distributed transmission, aggregation, storage, and access of events, data, and telemetry related to development projects, while giving individual developers fine-granularity control over the privacy of the data collected. To this end, the concept adapts an existing collaborative development and measurement infrastructure, the Context-aware Software Engineering Environment Event-driven framework (CoSEEEK) to support cloud-based event aggregation capabilities. The results and discussion show its practicality and technical feasibility while presenting performance tradeoffs for different cloud configurations. The concept enables infrastructural support for privacy, trust, and transparency within development teams, and could also support compliance with national privacy regulations in such dynamic and potentially global collaborative environments.**

Keywords - cloud-based software engineering environments, cloud-based software development collaboration, software project telemetry, privacy, security, trust.

I. INTRODUCTION

This article extends previous work in [1]. Global software development (GSD) [2] is increasingly taking advantage of cloud-based software applications and services [3] and realizing its collaboration potential. Data acquired and utilized during the software development and maintenance lifecycle is no longer necessarily locally controlled or even contained within an organization, but may be spread globally among various cloud providers with the acquired data retained indefinitely. Tools-as-a-Service (TaaS) [4] and cloud mashups will enable powerful new applications that utilize the acquired SE data [5]. And while the technical landscape is changing, the corporate landscape is also. A 2005 survey of American corporations conducted by the American Management Association showed that 76% monitored employee Internet connections, 50% stored and reviewed employee computer files, and 55% retained and reviewed email messages, with a rapidly increasing trend [6].

The ability to measure and minutely observe and track software developers during their work is becoming technically and economically viable to employers, managers, colleagues, virtual teams, and other entities. While metrics can be useful and provide a basis for improvements, be it at the organizational level (e.g., the CMMI Measurement and Analysis process area [7]), at the project level via automated software project telemetry (e.g., [8]), or for personal improvement (e.g., Personal Software Process [9], [10]). Unintended effects and abuse are also possible, such as [11] and [12], misuse of publicized information, misuse by competitors, mobbing, etc. While software services and apps developed by vendors for public customers typically attend to user privacy due to their longevity, mass accessibility, and regulatory and legal scrutiny, relatively little attention has been paid to the privacy needs of software developers, an estimated 17 million worldwide [13].

Consequently, privacy is becoming a looming concern for software developers that faces unique technical challenges that affect their collaboration. These challenges include: a highly dynamic technical environment typically at the forefront of software technology and paradigms (e.g., new languages, compilers, or platforms); diverse tools (for instance, [4] alone identifies 384); heterogeneous project-specific tool chains (e.g., application lifecycle management, version control systems, build tools, integrated development environments, etc.). Additionally, because development environments are often project-centric (unique and perhaps short-lived undertakings), the extra hassle and overhead for addressing developer privacy may seem to be an unnecessary hindrance to project progress and thus not be addressed at the management level. When multinational coordination (e.g., offshoring) is involved, multiple regulatory issues may apply and add to the complexity, etc. Developers may thus have little leverage and currently few technical options or suggestions for having their concerns addressed. Any privacy options should thus be economical and practically feasible, yet due to the dynamic technological nature of collaborative development environments (CDEs), standardization is unlikely or will be highly challenging.

To enable collaboration, the trust climate plays a vital role in the success of virtual and distributed teams [14], and trust and transparency are considered vital values for effective teams and collaboration [15][16]. Where trust exists (consider Theory Y [17]), collected data can be utilized collaboratively to enhance team performance [18], for instance by utilizing event data to coordinate and trigger

actions and to provide insights, whereas where data is misused as an instrument of power, monitoring, or controlling (consider Theory X [17]), individuals require mechanisms for protection. Since the technical development infrastructure cannot know a priori what trust situation exists between some spectrum of complete trust to complete distrust, infrastructural mechanisms should support collaboration within some spectrum, while allowing the individuals and organizations to adapt their level of data transparency to the changing trust situation. Not all actors involved may have an issue with metric collection, while those who favor complete transparency may presume that those voicing privacy issues may seem to be "hiding something".

Privacy is control over the extent, timing, and circumstances of sharing oneself. Cloud service users currently have few personal infrastructural mechanisms for retaining and controlling their own personal data. Diverse privacy regulations are applicable within various geographic realms of authority. Various (overlapping) (multi-)national laws and regulations may apply to such (global) collaborative cloud contexts. For instance, Germany has a Federal Data Protection Act, the European Union has a Data Protection Directive 95/46/EC, and within the United States, various states each have their own internet privacy laws. Many privacy and security principles are typically involved, including notice, consent, disclosure, security, earmarking, data avoidance, data economy, etc. Various challenges for security and privacy in cloud environments remain [19][20]. In the interim, pragmatic infrastructural approaches are needed to deal with the issues in some way.

As in the initial paper [1] on which this extended version is based, the contribution includes elucidating the requirements and describing a solution concept for pragmatically addressing various privacy and security concerns in cloud-based dynamic heterogeneous CDEs. The solution is based on service layering, introduces distributed *cloud-based datasteading* for individuals, and mediates trust via brokers. Its technical feasibility and performance tradeoffs are investigated in an initial case study. Additional contributions of this extended version include a discussion, details, and evaluation of an aggregator implementation supporting push-based event collection from personal datasteads.

This paper is organized as follows: the next section describes the assumptions and requirements for a solution, while Section III describes related work. In Section IV, the solution concept is introduced, with the following section providing details of a technical implementation based on the concept. Evaluation results are presented in Section VI. Section VII provides a discussion, which is then followed by a conclusion and description of future work.

II. REQUIREMENTS

The following requirements, assumptions, or constraints (denoted by the prefix *R: in italics*) were elicited from the primary problems, goals, and challenges introduced in the preceding section, and are considered to be generally applicable for any conceptual solution. They are summarized here to highlight key considerations in the solution concept.

Multi-cloud configurability (*R:MultCld*): in view of GSD, inter-organizational collaboration, and the long-term nature and scale of certain development projects, any solution should support private clouds (*R:PrivCld*), public clouds (*R:PubCld*), and community clouds (*R:CmtyCld*) for a wide array of deployment options.

Cloud portability or provider-specific cloud API independence (*R:CldPort*) should be supported to avoid cloud provider lock-in and allow wider adoption and applicability. Development teams tend to want choices in their tooling and infrastructure to optimize and tailor their project or situation based on costs or risks (business, quality-of-service, potential espionage risks, etc.). If this is challenging because cloud vendors do not want to change or agree to some common interoperability standard, adaptation techniques such as bridging, brokers, or mediators could be used to support common infrastructure functionality.

Cloud compatibility (*R:CldCmpat*) with current public cloud provider and private cloud APIs and services should be supported. This entails avoiding exotic requirements for special configurations that would constrain its practical usage, such as refraining from special hardware requirements such as the Trusted Platform Module (TPM), or obtuse software languages, platforms, operating systems, or communication mechanisms that, while perhaps increasing privacy or security to some degree, might nevertheless hinder overall adoption of such an approach because such configurations require too much effort or become too unwieldy or difficult to implement and maintain.

Single tenancy (*R:1Tenant*) in the personal (developer's) cloud should be supported to reduce risk (e.g., to avoid a misconfiguration compromising a much larger set of tenants simultaneously) and to avoid access by organizational administrators, which can involve an additional trust issue beyond the project level.

Disclosure (*R:Dsclsr*): three fundamental levels of disclosure shall be supported: non-disclosure, anonymized disclosure, and personally-identifiable disclosure to specific aggregators. This allows the developer to adapt the disclosure of events and data to the trust situation of a specific project or group.

Sensor Privacy (*R:SnsPriv*): It is assumed that any client-side and server-side sensors, (e.g., version control system sensors) distribute personally-identifiable events according to a privacy concept, or are at least configured in such a way that they only transmit their events securely directly to a single datastead.

Entity-level privacy control (*R:EntityCtrl*): the granularity of privacy is controllable by the entity involved or affected, be it persons, teams, organizations, projects, etc., and flows from bottom-up (from persons to teams) and across for similar levels (e.g., between teams or between organizations). Top-down controls can only restrict privacy, e.g., in the case where organizations no longer trust each other (perhaps due to legal action), they cannot forcibly increase the disclosure levels of lower entities.

Restrict network access (*R:R:PrivNtwk*) to collaboration participants only, e.g., via Virtual Private Networks (VPN), to reduce the accessibility of the communications to the collaborators only. It may also be useful even within a larger organization's intranet to reduce accidental leakage risk.

Secure communication (*R:SecComm*) can be used to protect internal or external data transmission. This may be considered useful even within a VPN for retaining personal privacy.

Basic security mechanisms (*R:BscSec*): this specifies the reliance on widely-available off-the-shelf security mechanisms (e.g., HTTPS), without any dependence on specialized or exotic hardware or software security platforms (e.g., Trusted Platform Module) or research-stage mechanisms that would constrain its practicality.

Encryption (*R:Encrypt*) can be used to protect data accessibility and storage.

Trusted code implementation (*R:TrstCd*): Open source and/or independent code audits together with secure distribution mechanisms (e.g., via digital signatures from a trusted website) provide assurance that the code implementation can be trusted.

Remote runtime *code integrity verification* (*R:Intgrty*) should be supported to allow agents (e.g., automated temporally random auditing requests or manually initiated user requests) to detect any tampering with the implementation, sensors, configuration, or the compromise of any privacy safeguards.

It is generally assumed that the environment and culture within an organization and between organizations is fundamentally one of mutual respect, benefit, and trust, with appropriate IT policies that reflect this, and that explicit surveillance and undermining tactics, tools, etc. are not tolerated or utilized to undermine employee personal privacy. In other words, this solution is not meant to address privacy and security at a hacker, professional, corporate, or espionage level, but rather to give developers choices in sharing their personal event and metric data with others in differing personal and project trust contexts and where they understand and know how that their collected data will be used to enhance productivity collaboratively. It has been said "you get what you measure," and, when applied to individuals, the repercussions could be greater the more exposure certain personal data has. This could result in (un)intentional manipulation or misinterpretation of the data out of context, in one direction to perhaps show off, and could negatively affect other hitherto positive interactions. E.g., if one measures individual programmer productivity and broadcasts this, then such desired behaviors as helping others or team or quality issues may be diminished or ignored. Other team members may very well be crucially supporting the development effort but not in ways currently being measured.

In summary, a primary tenet here is that organizations and teams want to support privacy freedom for individuals, that they support and value self-organizing teams, and that they do not wish to hinder electronic collaboration and communication. While together the aforementioned elucidated requirements are not intended to be sufficient or complete, they nevertheless provide a practical basis for considering and comparing solution concepts and can be useful for furthering discussion.

III. STATE OF THE ART

In the area of global software development, [4] discusses support for TaaS and [21] Software-as-a-service in collaborative situations. Neither go into detail on various privacy issues, nor is support for various aforementioned requirements, e.g., for individuals (*R:EntityCtrl*). Example industrial offers for cloud-based collaboration include Atlassian OnDemand and CollabNet CloudForge. Individual privacy control (*R:EntityCtrl*, *R:Dsclsr*) do not appear to be supported.

Work on more general multicloud collaboration includes [5], which similarly supports opportunistic collaboration without relying on cloud standardization based on the use of proxies. However, aspects such as (*R:BscSec*, *R:Intgrty, R:EntityCtrl*) were not considered and a technical implementation was not investigated.

Work in the area of standardization and reference architecture includes [22], which mentions privacy but fails to prescribe a solution. [23] lists various security and interoperability standards and their status, but their maturity and market penetration when considering (*R:MultCld*) and (*R:CldCmpat*) remain issues.

Various general cloud security mechanisms have been proposed. Privacy as a Service (PasS) [24] relies on secure cryptographic coprocessors to provide a trusted and isolated execution and data storage environment in the computing cloud. However, its dependency on hardware within cloud provider infrastructure hampers (*R:CldCmpat, R:CldPort,* and *R:BscSec*). Data protection as a service (DPaaS) [25] is intended to be a suite of security primitives that enforce data security and privacy and are offered by a cloud platform. Yet this would inhibit (*R:CldPort*). Other work such as [26] describe privacy-preserving fine-grained access control and key distribution mechanisms, but are not readily available for a pragmatic approach that is usable today (*R:BscSec*).

IV. SOLUTION CONCEPT

For a cloud-based context-aware collaboration system to have satisfactory utility, it will depend on some type of event and data collection and communication facilities. Thus, this foundational infrastructure should be equipped with basic trust and security mechanisms such that upper-level services like context-awareness and collaboration can ensue without impinging on privacy.

Thus, to provide a flexible solution for achieving privacy control in such environments, a primary principle in the solution concept is the application of the *Service Layer* design pattern [27] to provide a decoupling and separation of concerns as shown in Figure 1. The lower conceptual Event and Data Services Layer includes event and/or data services for an entity (person/team/organization), including acquisition, storage, retention, and dissemination, while the upper Collaboration and Tools Services Layer includes CDE and tool services. The upper layer services utilize lower layer data to provide collaboration, data sharing, analytics,

telemetry, contextual guidance, and other value-added services. Any single entity would have more limited privacy control mechanisms.

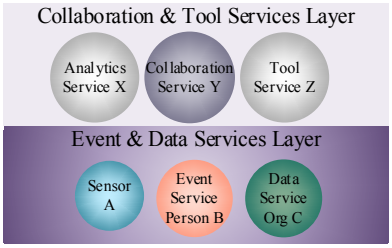


Figure 1. Services Layer Pattern.

A second solution principle is the introduction of a *datastead*, shown in Figure 2. Loosely analogous to the concept of homesteading or seasteading, it provides an entity with both a certain degree of data isolation and control for some area. In this case, some entity (be it an individual or some unit) manages and controls clearly delineated data resources in the cloud for which they have or receive responsibility and ownership rights. The technical implementation of a datastead can be in the form of a personal cloud in the case of an individual, or an area within a private cloud for an organization. It is thus clear to the individual or entity that they have complete control over personal (or entity) event and data storage that is kept separate under their personal (or entity) jurisdiction. Each datastead can pass data to one or more other datasteads (such as one belonging to a team) or directly to (usually one) community cloud where it can be processed and utilized to enhance collaboration. A configuration with successive, staged, or pipelined datasteads, while not required, can support the need for entity level privacy and disclosure control from the lowest levels to the highest levels in organizations (bottom-up). Community clouds may also successively pass data on to larger community clouds if desirable to the providing community. For instance, academic research communities could access and analyze this data for multiple projects.

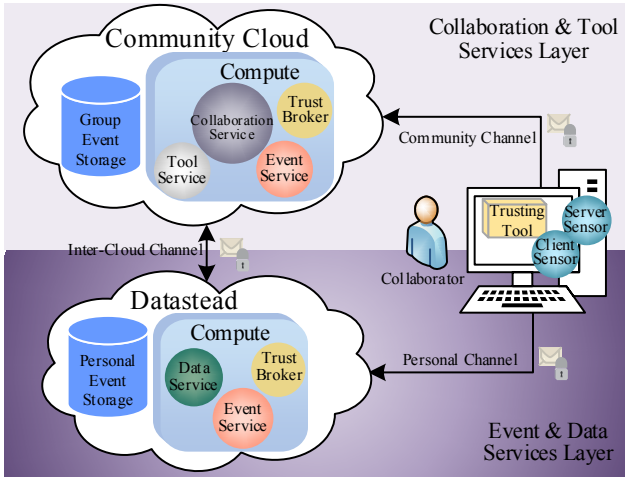


Figure 2. Generic Solution Concept.

The third principle is the inclusion of a Trust Broker that mediates between service and data access, acting as both a cloud service broker (for interoperability with various tools) and cloud security broker (for security) between layers. Akin to the Trusted Proxy pattern [28] and Policy Enforcement Point [29], it constrains access to protected resources and allows custom, finely-tuned policies to be enforced (*R:EntityCtrl*). Rules can be used to configure and distinguish/filter access by event types, timeframes, projects, etc. It provides secure communication mechanisms (*R:SecComm*) to authenticate and authorize data acquisition and data dissemination in the datastead, as well as interoperability mechanisms for various collaboration and tool services. Only client requests from preconfigured known addresses are accepted. A management interface to the Trust Broker provides the datastead owner with policy management capabilities. It also supports data anonymization on a per request basis if so configured. For secure storage, the Trust Broker encrypts (*R:Encrypt*) acquired events and data (Encrypted Storage pattern [28]) to prevent unauthorized access by administrators or intruders, and protects access to the encrypted storage typically on a single port (Single Access Point pattern [29]). The Trusting Broker supports runtime code integrity (*R:Intgrty*) via remote attestation, and a client, called the Trusting Tool, can be invoked periodically or based on certain events to ensure that the Trust Broker has not been tampered with.

As to transmission, a Personal Channel transmits events from sensors to the personal datastead. The Inter-Cloud Channel transmits personal or anonymized events to one or more Community Clouds. The Community Channel is optional and can be used, e.g., for impersonal sensors (e.g., team build server) or perhaps in special situations when duplication and parallel transmission of personal events for reliability or performance is desired and approved. Secure Channels and Secure Sessions [29] are used to protect the transmission between the sensors and the datastead (the Personal Channel), between sensors and the Community Cloud (Community Channel), as well as between the datastead and any collaboration and tool services (Inter-cloud Channel). For a community cloud, a VPN is used to limit network access to collaborators in the community only.

V. TECHNICAL IMPLEMENTATION

To determine the technical feasibility of the solution concept and provide a concrete case study, the solution concept was applied to an existing heterogeneous CDE called the Context-aware Software Engineering Environment Event-driven framework (CoSEEEK) [30], which had hitherto not incorporated privacy or security techniques. CoSEEEK's architecture and integrated technologies are shown in Figure 3. Its suitability is based on its portability (use of mainly Java and web-based languages), use of non-commercial technologies described below, its reliance on common distributed communication mechanisms such as RESTful web services, and its heterogeneous tool support. Additional technical details on CoSEEEK can be found in [31][32].

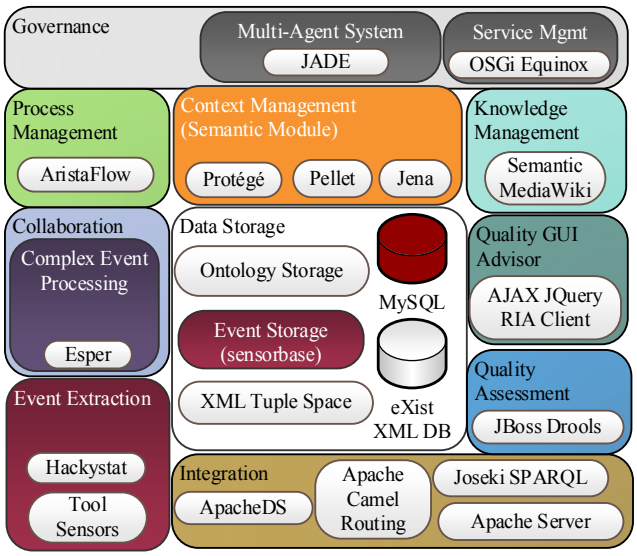


Figure 3. CoSEEEK Architecture (affected areas shown in red).

For event acquisition, CoSEEEK relies on the Hackystat framework [33] and its SE tool-based sensors (e.g., Ant, Eclipse, Visual Studio) for event extraction and event storage (shown in red in Figure 3). Hackystat does not currently provide extensive security and privacy mechanisms. For an insight, [34] briefly describes some of its security issues.

Service Layer Separation: the Hackystat-related elements (shown in red) were hereby separated into the Event and Data Services Layer and the remaining elements were placed in the Collaboration and Tools Services Layer.

Cloud configuration: To meet (*R:MultCld*, *R:CldCmpat*, *R:CldPort*), two different cloud platforms were utilized in isolation. To represent a public IaaS cloud provider configuration (*R:PubCld*), Amazon Web Services (AWS) was used, using Elastic Compute Cloud (EC2) for computing services, the Elastic Block Store (EBS) for storing configuration files and XML database, and the Relational Database Service (RDS) that holds the sensorbase.

To represent a private cloud (*R:PrivCld*) or community cloud (*R:CmtyCld*) deployment, OpenStack was used with Compute used for computing and Object Storage used in place of EBS storage. Since nothing directly equivalent to AWS RDS was available, we configured a Compute instance with Object Storage that contains a MySQL Server database and for single tenancy (*R:1Tenant*) one Compute instance per developer with access restricted to the developer.

Trust Broker: the Trust Broker supports (*R:Dsclsr*) was implemented in Java using a REST framework. An example of a query that can be sent is the following, specifying the project via the sensorbase_id, the timeframe, the sensor data type, the tool, and its uri source.

```
   GET
/trustbroker/sensordata/{sensorbase_id}?
startTime={startTime}&endTime={endTime}&
sdt_name={sdt_name}&tool={tool}
   &uriPatterns={uriPatterns}
```

Encryption of events (*R:Encrypt*) can be optionally configured. For encryption of arriving events and decryption of events on authenticated and authorized retrieval, Java's AES 128 and the SHA-256 hash algorithm were used (*R:BscSec*). One reason for encrypting the storage is that it provides an additional form of protection, should, e.g., a provider's agent or intruder gain access.

The measurement database, called sensorbase in Hackystat, required a few minor adaptations. For (*R:Dsclsr*) to support anonymization, the HACKYUSER table was extended to include an anonymization flag that is checked before responding, replacing a userid with anonymous. In order to support HTTPS connections, the sensorbase client (*R:SnsPriv*) was modified and rebuilt, requiring any sensors to utilize this modified jar file. HTTPS (*R:BscSec*) was used to secure all three communication channels (personal, community, and inter-cloud) (*R:SecComm*). Additional properties were added to indicate the location of the keystore. SSH was used to configure and manage each cloud. Security groups were used in both AWS and OpenStack.

A. Aggregator

To improve cloud-based performance, we adapted the solution concept from our initial paper to remove the ongoing querying of the datastead by the Trust Broker in the community cloud. Instead, a client-based push approach for event transmission was implemented. A blacklist within the client Trust Broker filters or anonymizes the types of events that are passed on and made available. The aggregation of the events now avoids polling the clients. For this, the client Trust Broker is responsible for tracking which events were already successfully transferred and which still need to be sent to any Aggregator in the Trust Broker on a Community Cloud.

The interaction between components for the transfer of events is shown in Figure 4. Sensors send their events on a push basis to the Hackystat sensorbase located in the datastead. The datastead Trust Broker periodically queries the sensorbase for events newer than its last transmission to any particular Aggregator. Once an event is fetched, it is checked against a blacklist to determine if it should be blocked or anonymized. Then the datastead Trust Broker pushes the event via its REST connection to the Aggregator within the Trust Broker residing on the Community Cloud, where it is persisted and can be processed by various upper-level services in CoSEEEK. This is repeated in a loop until the latest event has been sent.

There are valid arguments for maintaining either a whitelist or blacklist, depending on what standpoint one takes (send most data vs. send almost no data), the extent of the associated rules, as well as the consideration of what should happen if something is not specified. In the case of a blacklist where something is missing or was misconfigured, that would imply that events would slip through.

While a whitelist could have been used, for simplification of the implementation for demonstration purposes we chose to use a blacklist, since we presume that developers will more likely know exactly what sensors they want to block or anonymize (i.e., blacklist) from the community more than

they will care to know and manage all the diverse and possible sensors (whitelist).

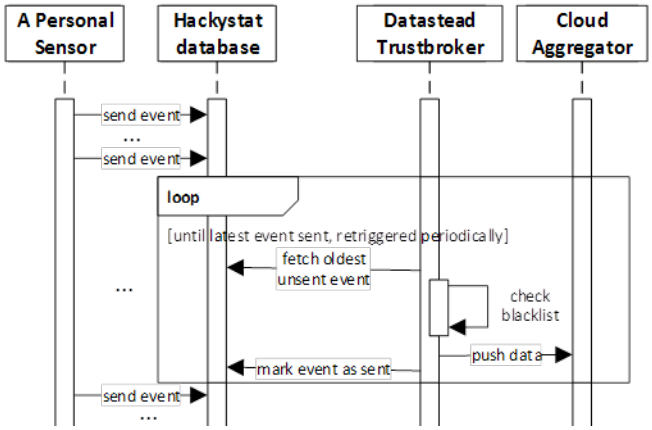


Figure 4. Sequence diagram showing push-based aggregation.

The file blacklist.xml specifies which events should be blocked or anonymized, whereby blocking is the default. The boolean tag `<Anonymize>` controls anonymization, but can be explicitly set to `false` as a kind of "unblock" to allow a specific event to be unblocked when most others are blocked.

If no events should be anonymized or blocked, then the list is empty as shown below.

```
<Blacklist>
</Blacklist>
```

By default, if a tool is listed all events from that tool sensor are blocked and remain in the datastead. For example, to block all events from the eclipse tool sensor, it would be specified as follows:

```
<Blacklist>
  <Tool>
    <Name>Eclipse</Name>
  </Tool>
</Blacklist>
```

To anonymize, for example, all events from the eclipse tool sensor, it would be specified as follows:

```
<Blacklist>
  <Tool>
    <Name>Eclipse</Name>
    <Anonymize>true</Anonymize>
  </Tool>
</Blacklist>
```

For example, to by default block all the other Eclipse tool events, but anonymize only the Eclipse File Open events, it would be specified as follows:

```
<Blacklist>
  <Tool>
    <Name>Eclipse</Name>
    <Property>
      <Anonymize>true</Anonymize>
      <Key>Subtype</Key>
      <Value>Open</Value>
    </Property>
  </Tool>
</Blacklist>
```

To anonymize all Eclipse events, and by default block the Eclipse File Open events, one specifies additional properties, as shown here:

```
<Blacklist>
  <Tool>
    <Name>Eclipse</Name>
    <Anonymize>true</Anonymize>
    <Property>
      <Key>Subtype</Key>
      <Value>Open</Value>
    </Property>
  </Tool>
</Blacklist>
```

To anonymize all Eclipse events except the File Open events, for example, the following would be specified:

```
<Blacklist>
  <Tool>
    <Name>Eclipse</Name>
    <Anonymize>true</Anonymize>
    <Property>
      <Anonymize>false</Anonymize>
      <Key>Subtype</Key>
      <Value>Open</Value>
    </Property>
  </Tool>
</Blacklist>
```

To manage what event to push, a simple event timestamp reference that is persisted tracks the last event successfully retrieved from the Hackystat sensorbase and that was either blocked due to the blacklist or transmitted to a specific Aggregator on a per event basis. Should an error during transmission occur, the client is responsible for retransmitting. Should the Aggregator become unavailable, the client will continue to retry rebuilding a connection until it succeeds in pushing the events not yet successfully transmitted in the order of their occurrence (timestamp). No separate queue is maintained and all events are stored in the sensorbase.

B. Remote Attestation

To implement remote attestation, on the client-side, a user configures the Trusting Tool with the expected checksum value (provided, e.g., by the admin or a trusted website), version, and the interval for rechecking. On the

service side, a REST interface sensorbase/checksum was added that loads the local adapted sensorbase.jar file, computes the SHA-256 hash value using `java.security.MessageDigest`, and returns this value and the sensorbase version to the Trusting Tool. While not foolproof, since any unauthorized access on the server or client could allow spoofing, it provides an additional level of confidence. Various stronger jar file tampering technologies could be employed if needed, such as componio JarCryp bytecode encryption.

VI. EVALUATION

The case study evaluated the technical feasibility of the concept based on the technical implementation. However, security and privacy are highly contextually dependent on the expectations, requirements, environment, risks, policies, training, available attack mechanisms, implementation details (bugs), configuration settings, etc. Therefore, making a comprehensive formal assessment in this area is difficult. So the assumption is made that the prescribed privacy and security mechanisms suffice or are balanced for current developer needs in developer settings.

Since CoSEEEK is a reactive system, the ability to respond adequately to contextual changes via events is dependent primarily on event latency. Cloud networking, additional network security mechanisms, and the additional delay incurred by inserting datastead nodes could negatively affect responsiveness, and thus this infrastructural level of event latency was the primary focus of the evaluation.

To elaborate, as CoSEEEK is a process-centered software engineering environment, any events that arrive too late to be contextually relevant can cause CoSEEEK-triggered actions or responses to be irrelevant and thus ignored. Developers also tend to be impatient when a guidance system is not providing relevant and applicable guidance for the context when expected, and they will continue on without it and perhaps begin to ignore it. As to event volume, events generated by any single developer's actions are typically sporadic and not highly voluminous. If a sensor is overly vociferous in relation to the amount of developer activity, it can typically be configured to eliminate redundant events or to summarize events. If this capability is not built in, a complex event processor (e.g., Esper) can be utilized to reduce the load on the network and aggregator in larger project environments.

A subjective evaluation by developers in an industrial setting was considered but not feasible at this time due to resource and schedule constraints, and is included in future work.

A. Security Overheads

To determine security overheads, the Client PC (for use by a developer) has an i5-2410M (2.3-2.9 GHz) dual core CPU and 6GB RAM with 32-bit Windows XP SP3. The network consists of gigabit Ethernet and two 1 Gbit connections from the university campus in Germany to the Internet Provider.

Representative for a private (*R:PrivCld*) or community cloud where a datastead could also be placed, the OpenStack configuration (OSCfg) consisted of a local intranet server with an i5-650 (3.2-3.4GHz) dual core CPU, 8GB RAM, and 64-bit Ubuntu Server 12.04. The OpenStack Cloud Essex Release was installed on the Server via DevStack and the Compute instances also ran Ubuntu Server 12.04. MySQL v. 5.5.24 was used for Hackystat sensorbase storage in a Compute instance.

As a public cloud provider (*R:PubCld*) representative, a free AWS configuration (AWSCfg) was chosen. It consisted of t1.micro EC2 instance types located in US-EAST-1d (Virginia) with 613 MiB memory, up to 2 EC2 units (for short periodic bursts) with low I/O performance running 64-bit Ubuntu Server 12.04. MySQL v. 5.5.27 was used for the Hackystat sensorbase storage in AWS RDS.

Common software included Hackystat 8.4 with the Noelios Restlet Engine 1.1.5 and JDK 1.6.

Typical network usage scenarios were considered, thus no optimizations were applied to any configurations nor was an artificially quiet network state created. All results are the average of 10 repeated measurements (with one exception noted below). A secure configuration denotes using the TrustBroker via HTTPS (*R:SecComm*) with encrypted storage (*R:Encrypt*), and an insecure configuration means HTTP without a TrustBroker. VPN (*R:R:PrivNtwk*) overheads were not measured.

To determine delays from the client to the datastead in cloud variants, on the client PC the Ant build tool was invoked, causing the Hackystat Ant sensor to send one XML event to the Server (a write in the remote sensorbase) consisting of 235 bytes of event data plus 73 bytes of network protocol overhead. The measured latency values are shown in Table I and Figure 5.

TABLE I. LATENCY (IN MS) FOR SENDING AN EVENT (235 BYTES) FROM THE CLIENT PC TO THE SERVER SENSORBASE

Insecure Private Cloud (ms)	Secure Private Cloud (ms)	Secure AWS Cloud (ms)
214	389	608

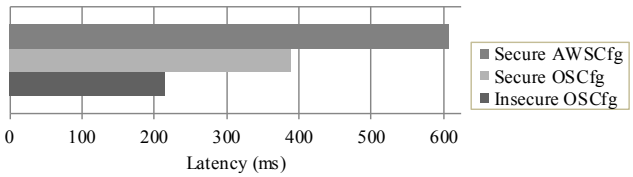


Figure 5. Latency (in ms) for sending an event (235 bytes) from the client PC to the server sensorbase.

Once events are in the datastead, then latencies incurred between cloud computing instances are of interest, since collaboration services or tool services will require this data. The measured values are shown in Table II and grouped by security mechanisms in Figure 6.

A grouping by cloud type is shown in Figure 7. For AWSCfg, a single query for 67 events (15818 bytes) between two EC2 instances took 78 ms on average via HTTP and 84 ms over HTTPS. In a secure configuration the retrieval took 347 ms. For OSCfg between two Compute

instances, a single query took 38 ms to return 22 events (5243 bytes). Note that HTTP insecure reads in the private cloud had two anomaly values (178 and 210 ms) that would have changed the average from 38 to 69, and were also far larger than any secure value measurements. Thus, these two measurement values were removed, and the average created from the remaining 8 values. These large latencies could perhaps be attributed to a network, disk, operating system, or OpenStack related issue. Continuing with the measurements with 39 events (9238 bytes), HTTPS requests took 60 ms while in the secure configuration it averaged 61ms. The overhead of the privacy approach is the addition of SSL, brokering a second SSL connection, and encryption. For the OSCfg, the difference of TrustBroker and decryption showed on average only a 1ms difference to that with purely SSL. One explanation could be that the extra overhead is minimal compared to the data transfer delays between OpenStack instances, but further investigation of OpenStack internals and performance profiling would be needed to clarify this.

TABLE II. PRIVATE VS. PUBLIC CLOUD INTER-COMPUTING INSTANCE QUERY LATENCIES (IN MS)

	Inter-OSCfg Latency (ms)	Inter-AWSCfg Latency (ms)
Insecure	38	78
HTTPS	60	84
Secure	61	347

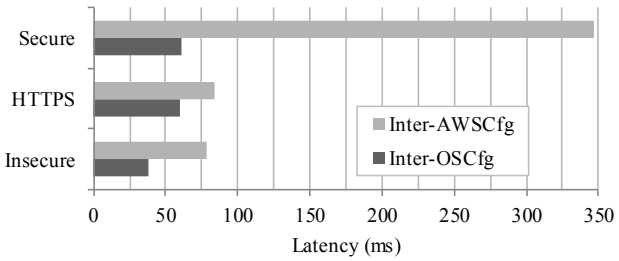


Figure 6. Private vs. public cloud inter-computing instance query latencies grouped by security (in ms).

Based on the results shown in the above figures, the use of the secure configuration of the OSCfg within a private or even a community cloud setting would appear to have acceptable performance overhead for cloud-centric collaborative development work, and distributed retrieval from datasteads is viable for responding to changes in the collaborative situation. On the other hand, the use of the secure configuration in the public cloud (AWSCfg), as shown in this perhaps worst case as a free offshore minimal public cloud setting, incurs substantially higher network latencies. Obviously, choosing geographically close locations when possible is recommended. Also, provisioning sufficient computing and I/O resources support to deal with the additional inter-cloud and security mechanism overheads would also reduce such lags in public cloud configurations. Optimizing this area could yield performance improvements but may incur additional financial costs.

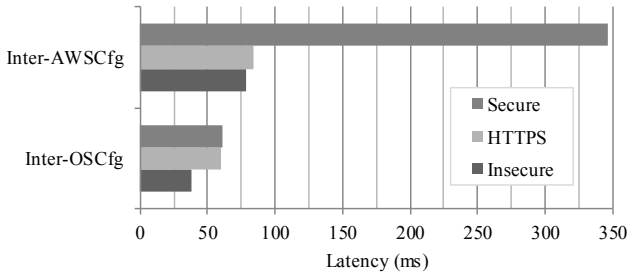


Figure 7. Inter-cloud query latency grouped by cloud type for different degrees of security (in ms).

To determine the remote attestation overhead, the Trusting Tool was measured on the PC using the AWSCfg over SSL. The average request-response latency was 702 ms. On the server, this involved loading and calculating the SHA-256 hash value for the 5.5 MB large sensorbase.jar file. Thus, the attestation mechanism of the remote cloud instance could be configured to be automatically invoked periodically by client-side sensors at regular intervals in a separate thread or process so as not to interfere with other network communication.

B. Aggregator Performance

In order to remove the query of datasteads by the Trust Broker Aggregator, the implementation was changed to a client-based push approach as mentioned in the previous section. The aggregator push implementation was measured separately to determine its performance and adequacy. For this a client PC served as the datastead. The following hardware setup was used for these measurements: the client was a Lenovo ThinkPad X201T with 2GHz Intel Core i7 L620 and 4GB RAM. The local server consisted of a PC with a 3.30GHz Intel Core i3-3220 CPU with 4GB RAM. Amazon AWS T2.micro consisted of 1 VCPU with 1GB RAM. All used a 64-Bit Ubuntu version 14.04. The network connection consisted of a 1 Gbit LAN between the PC and server locally and 2.5Mbit upstream and 50Mbit downstream to the internet provider. HTTP with REST was used for these measurements using Jersey 2.10 and the Java Runtime Environment 7.

Since the filtering of events will not consume significant wall clock time in comparison to the network aggregation, event filtering and anonymization were disabled. 128 events were pushed to the Aggregator from the client. Since events are not likely to be excessively large, events of 256 and 512 bytes in total length were used for comparison. The results are shown in Table III and Figure 8.

No significant latency differences due to a larger event size were detected on the local network. This can be explained in that the primary overheads involved are not related to content analysis or processing of data within the packets or events since this was not performed, and that the high network transmission rate available made the additional payload insignificant.

The latency durations indicates that potential exists here for performance optimization, but due to time and resource constraints a more thorough analysis of these initial results

using CPU profiling and network sniffers could not be performed.

TABLE III. LOCAL VS. PUBLIC CLOUD NETWORK AGGREGATION LATENCIES (IN MS) FOR 128 EVENTS

Event size (bytes)	Local Aggregator Duration (ms)	AWS Aggregator Duration (ms)
256	7528	10448
512	7527	10617

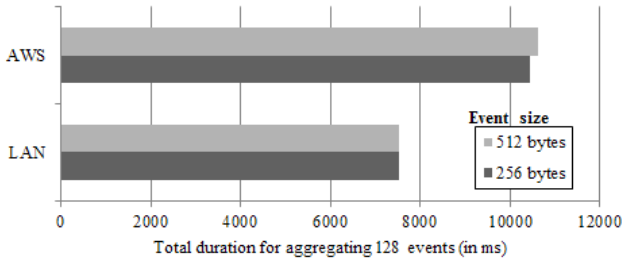


Figure 8. Aggregation network latencies (in ms) in local vs. public cloud settings for 128 events.

In summary, the evaluation showed that network latencies incurred by the solution concept are most likely insignificant for collaboration in PrC settings, but that security overheads in global PuC settings may require optimization attention to minimize their effects.

VII. DISCUSSION

Telemetry and metrics play a vital role in providing a data basis for assessing areas for improvement, for benchmarking against other organizations or projects, as a basis for root cause analysis, and for determining the effects of any improvement initiatives.

When the trust environment in an organization or project is healthy, then the sharing of event data and associated metrics can be used to support tighter collaboration, streamline interactions, and be used in retrospectives for analysis to support and verify improvements and best practices. However, when the trust environment is degraded or non-existent, then a forced sharing of detailed personal-level data may result in additional inefficiencies due to psychological or motivational effects, circumvention or abuse of such a system, or adapting behaviors or sensors to intentionally providing misleading data to make certain people look good and/or others look worse.

The Tuckman model [35] for the development of small groups may be useful to illustrate the difficulties as teams transition in their interactions and the associated change in the trust among members, from forming to storming, norming, performing, and then adjourning. The concept in this paper can adjust for increasing trust, from blocked to anonymous to personalized events. Should the trust situation decline, it can support adjustments in the policies from that point on for new events. Any events already disclosed will however remain so unless the community cloud is manually cleansed by an administrator.

If we look beyond software developers and at the broader picture of employees in organizations that intentionally monitor their employees, they are then likely to utilize surveillance products that were intentionally built for this purpose and will likely not explicitly involve their employees. Addressing privacy in such situations is beyond the technical scope of the concept and approach in this paper and will likely need to rely on regulatory mechanisms to balance the rights and responsibilities of the parties involved.

However, due to the dynamics of projects and development environments, and the freedom and influence or empowerment developers often have, software developers are in a unique position to influence the use of measurements for team improvement while balancing the amount of transparency and personal measurement to the shifting trust environment. Personal empowerment over personal measurement data may allow developers to embrace the adoption and inclusion of personal metrics and enhance the productivity of teams without the negative impacts of forced submission to measurement collection. The adoption in open source projects, for example, would allow deeper analysis and understanding, even if the metrics and events were anonymous. In the larger scheme of things, this could provide the software engineering research community with valuable additional data for (meta-)analysis and improvements in the hitherto semi-inaccessible data collection area associated with software development processes.

The approach in this paper intended to provide personal control mechanisms to developers to deal with privacy and trust reservations of developers towards the integration of sensors in their environments needed by collaboration and telemetry systems similar to CoSEEEK.

VIII. CONCLUSION AND FUTURE WORK

To address security and privacy in collaborative cloud development, this paper presented a practical concept with entity-level control of non-, anonymized-, and personally-identifiable disclosure for multiple cloud configurations. It can further both collaboration and trust by giving individuals transparency and control and allowing them to adjust disclosure to the changing trust situation. The paper contributes a practical basis for illustrating issues, eliciting awareness, community discussion, and may increase self-regulation and infrastructural privacy offerings. Organizations adopting such a privacy infrastructure show that they value and trust their employees, enabling them to reap mutual trust rewards. Also, one could envision, for instance, that an audited "we don't spy here" seal might help attract and retain developers for certain organizations.

The evaluation showed its technical feasibility and practicality, requiring only minimal adaptation of the CoSEEEK CDE. The Trust Broker enables fine granularity access control to personal data. Performance was sufficient in private cloud configurations, while public cloud configurations using additional security and privacy mechanisms may require optimization to ensure fluid collaboration situational response. The push-based aggregation supports black-list filtering and anonymization

on a fine-grained event basis and better supports aggregating events when many clients are involved. The current implementation relies on the clients to avoid retransmission of events. Should this not suffice in practice, the Aggregator could be adapted to ensure that duplicates are not stored, e.g., by using a hash value or checksum for each received event, and any new events compared with all previous event hash values, although this would add some additional overhead.

Limitations and risks include: extending privacy/trust support within and across collaboration layer tools, non-detection/discovery of (un)intentionally unspecified/hidden sensors, data manipulation risk by datastead owners themselves, and provider-side access or manipulation risk. In the case of trust issues with the service provider, building your own datastead cloud server site could be considered. A concept for reliably maintaining and updating the software across the various datasteads in a trustworthy and efficient manner, with or without manual intervention by the datastead owner, should also be considered. Perhaps the updates should require some certification and could then be performed automatically if desired by the owning entity. In the end, developers will likely prefer low hassle solutions that still provide adequate privacy transparency and controls.

Future work includes an industrial field study, the inclusion of various data provenance and data integrity mechanisms to mitigate manipulation risk, and the investigation of enhanced remote attestation mechanisms. In the face of shifting privacy norms, infrastructural support for data confidentiality is needed to limit disclosure of distribution data beyond its original intent, like lifetime constraints, transitivity bounds, and claims-based access [36]. One challenge here is to deal with annulment or revocation of data already shared in the past when the trust situation degrades. Since service privacy is also a broader issue, development and adoption of global industry service privacy standards combined with independent privacy audits involving all service layers would enhance the trust of cloud-based data acquisition and service offerings.

ACKNOWLEDGMENT

The author wishes to acknowledge Roman Pisarew and Jürgen Drotleff and for their assistance with the concept, implementation, and measurements.

REFERENCES

[1] R. Oberhauser, "Towards cloud-based collaborative software development: A developer-centric concept for managing privacy, security, and trust," Proceedings of the Eighth International Conference on Software Engineering Advances (ICSEA 2013), pp. 533-538.

[2] S. Hashmi et al., "Using the cloud to facilitate global software development challenges," in Proceedings of the Sixth IEEE International Conference on Global Software Engineering Workshop (ICGSEW), IEEE, 2011, pp. 70-77.

[3] R. Grossman, "The case for cloud computing," IT professional, 11(2), 2009, pp. 23-27.

[4] M. Chauhan and M. Babar, "Cloud infrastructure for providing tools as a service: Quality attributes and potential solutions," in Proceedings of the WICSA/ECSA 2012 Companion Volume, ACM, 2012, pp. 5-13.

[5] M. Singhal et al., "Collaboration in multicloud computing environments: Framework and security issues," Computer, 46(2), IEEE Computer Society, New York, 2013, pp. 76-84.

[6] G. Nord, T. McCubbins, and J. Nord, "E-monitoring in the workplace: Privacy, legislation, and surveillance software," Communications of the ACM, 49(8), 2006, pp. 72-77.

[7] C. Team, "CMMI for development, version 1.3, improving processes for developing better products and services," no. CMU/SEI-2010-TR-033, Software Engineering Institute, 2010.

[8] P. Johnson, H. Kou, M. Paulding, Q. Zhang, A. Kagawa, and T. Yamashita, "Improving software development management through software project telemetry," IEEE Software, 22(4), 2005, pp. 76-85.

[9] W. Humphrey, "The personal software process: Status and trends," IEEE Software, 17(6), 2000, pp. 71-75.

[10] P. Johnson et al., "Beyond the personal software process: Metrics collection and analysis for the differently disciplined," Proceedings of the 25th international Conference on Software Engineering, IEEE Computer Society, May 2003, pp. 641-646.

[11] R. Irving, C. Higgins, and F. Safayeni, "Computerized performance monitoring systems: Use and abuse," Communications of the ACM, 29(8), 1986, pp. 794-801.

[12] J. Stanton, "Reactions to employee performance monitoring: Framework, review, and research directions," Human Performance, 13(1), 2000, pp. 85-113.

[13] M. Parsons, "The challenge of multicore: A brief history of a brick wall," EPCC News, Issue 65, University of Edinburgh, 2009, p. 4.

[14] T. Brahm and F. Kunze, "The role of trust climate in virtual teams," Journal of Managerial Psychology, 27(6), 2012, pp. 595-614.

[15] A. Costa, R. Roe, and T. Taillieu, "Trust within teams: The relation with performance effectiveness," European journal of work and organizational psychology, 10(3), 2001, pp. 225-244.

[16] T. DeMarco and T. Lister, Peopleware. Dorset House, 1987.

[17] D. McGregor, The Human Side of Enterprise. McGrawHill, New York, 1960.

[18] B. Al-Ani and D. Redmiles, "Trust in distributed teams: Support through continuous coordination," IEEE Software, IEEE Computer Society, 26(6), 2009, pp. 35-40.

[19] P. Louridas, "Up in the air: Moving your applications to the cloud," IEEE Software, 27(4), IEEE Computer Society, New York, 2010, pp. 6-11.

[20] H. Takabi, J. Joshi, and G. Ahn, "Security and privacy challenges in cloud computing environments," IEEE Security & Privacy, IEEE Computer Society, 8(6), 2010, 24-31.

[21] R. Martignoni, "Global sourcing of software development-a review of tools and services," In Fourth IEEE International Conference on Global Software Engineering (ICGSE 2009), IEEE Computer Society, 2009, pp. 303-308.

[22] F. Liu et al., NIST Cloud Computing Reference Architecture. NIST Special Publication, 500, 292, 2011.

[23] M. Hogan, F. Liu, A. Sokol, and J. Tong, NIST Cloud Computing Standards Roadmap. NIST Special Publication, 35, 2011.

[24] W. Itani, A. Kayssi, and A. Chehab, "Privacy as a service: Privacy-aware data storage and processing in cloud computing architectures," In Eighth IEEE International Conf. on Dependable, Autonomic and Secure Computing (DASC'09), IEEE Computer Society, 2009, pp. 711-716.

[25] D. Song, E. Shi, I. Fischer, and U. Shankar, "Cloud data protection for the masses, " Computer, 45(1), 2012, pp. 39-45.

[26] M. Nabeel and E. Bertino, "Privacy-preserving fine-grained access control in public clouds," Data Engineering, 21, 2012.

[27] T. Erl, SOA Design Patterns. Pearson Education PTR, 2008.

[28] D. Kienzle, M. Elder, D. Tyree, and J. Edwards-Hewitt, Security Patterns Repository Version 1.0. DARPA, 2002.

[29] M. Schumacher, E. Fernandez-Buglioni, D. Hybertson, F. Buschmann, and P. Sommerlad, Security Patterns: Integrating Security and Systems Engineering. Wiley, 2006.

[30] G. Grambow, R. Oberhauser, and M. Reichert, "Enabling automatic process-aware collaboration support in software engineering projects," in Software and Data Technologies, Springer, Berlin Heidelberg, 2013, pp. 73-88.

[31] R. Oberhauser, "Leveraging semantic web computing for context-aware software engineering environments," Semantic Web, Gang Wu (ed.), In-Tech, Austria, 2010.

[32] G. Grambow, R. Oberhauser, and M. Reichert, "Contextually injecting quality measures into software engineering processes," the International Journal On Advances in Software, ISSN 1942-2628, vol. 4, no. 1 & 2, 2011, pp. 76-99.

[33] P. Johnson, "Requirement and design trade-offs in Hackystat: An in-process software engineering measurement and analysis system," Proc. First Intl. Symposium on Empirical Software Engineering and Measurement, 2007, pp. 81-90.

[34] P. Johnson, C. Moore, J. Miglani, and S. Zhen, Hackystat Design Notes. 2001.

[35] B. Tuckman, "Developmental sequence in small groups," Psychological bulletin, 63(6), 1965, p. 384.

[36] D. Reed, D. Gannon, and J. Larus, "Imagining the future: Thoughts on computing," Computer, 45(1), 2012, pp. 25-30.

A Social Approach for Natural Language Query to the Web of Data

Takahiro Kawamura
Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
e-mail: kawamura@ohsuga.is.uec.ac.jp

Akihiko Ohsuga
Graduate School of Information Systems
University of Electro-Communications
Tokyo, Japan
e-mail: akihiko@ohsuga.is.uec.ac.jp

***Abstract*—The 'Web of Data' aims to enable people to share structured data as easily as they can share documents on the Web. Accordingly, a search engine for the data is becoming important for promoting data-intensive services. However, the data are represented and interlinked by a triple format in the Resource Description Framework, and thus full-text search is unsuitable for structured data, and formal query languages are difficult for ordinary users. Therefore, we propose a query answering system, which accepts natural language queries in Japanese, translates them to triples and sends formal queries to the 'Web of Data'. The proposed system is implemented on a mobile phone, and evaluated in the context of gardening advice for people. The use is within a social system, and combines user feedback and user context information obtained by sensors, in order to improve search accuracy for open-schema mapping and data acquisition. Then, we confirmed that the search accuracy of 18.2% was improved by the user feedback, and the useful triples have been increased 3.4 times by the context information.**

Keywords–Web of Data; Linked Open Data; Query Answering; User Context; Plant.

I. INTRODUCTION

The 'Web of Data' aims to enable people to share structured data as easily as they can share documents on the Web, and is currently attracting attention because it is expected to enable the creation of innovative service businesses, mainly in the areas of government, bioscience, and smart city projects. To promote the application of the data in a large number of services, it would be helpful to have a search engine for the 'Web of Data'. Given that the data format is described as the Linked Open Data (LOD) in Resource Description Framework (RDF) (as of December 2012, RDF is a candidate for the standard format of the open data at the Japanese Ministry of Internal Affairs and Communications), full-text search is unsuitable for data fragments in the structured data. Moreover, it is difficult for ordinary users to perform a formal query using SPARQL Protocol and RDF Query Language (SPARQL). Therefore, we propose a query answering system for matching triples extracted from the user query sentence to triples in the 'Web of Data'. For instance, the system accepts natural language queries in Japanese, translates them to RDF triples $< subject, verb, object >$ and sends SPARQL queries to the RDF DB. In this paper, we focus on the mapping of query sentences to open-schema data and data acquisition, and then attempt to improve search accuracy based on user feedback and to acquire new data from user context information. We evaluate these points using 'Flower Voice', which is an application of the query answering system implemented on a mobile phone for assisting users with their gardening. Finally, we indicate the results of performance evaluations.

The remainder of this paper is organized as follows. Section II presents several examples of related work, and Section III describes problems with and approaches to realizing the query answering system for LOD. Then, Section IV proposes an application of this software, Flower Voice [1], which is a smartphone tool for searching for information and for logging gardening activity. Finally, we conclude by referring to future work in Section V.

II. RELATED WORK

In research on query answering (QA) systems and databases, many attempts have been made to automatically translate from natural language queries to formal languages, such as Structured Query Language (SQL) and SPARQL, in order to facilitate the understanding of ordinary users and even database (DB) experts. Research also exists on outputting the queries and results into natural language sentences [2][3]. Although, as we pointed out in Section I, it is difficult to apply full-text search to data fragments, there has been research on converting a keyword list to a logical query [4][5][6].

In this section, we classify research on QA systems that translate natural sentences into queries, which are classified into two categories based on whether a deep or shallow linguistic analysis is needed.

One system that requires deep linguistic analysis is ORAKEL [7][8]. It first translates a natural sentence into a syntax tree using Lexicalized Tree Adjoining Grammars, and then converts it to F-logic or SPARQL. Although it is able to translate while retaining a high degree of expressiveness, it also requires the original sentence to be precise and regular. Wendt et al. [9] considers a QA system together with the design of a target ontology mainly for event information, and features handling of temporality and N-ary during the syntax tree creation. It assigns the words of the sentence to slots in a constraint called a *semantic description* defined by the ontology, and finally converts the semantic description to SPARQL recursively. In terms of a natural language QA system for ordinary users applicable to LOD, however, these approaches are problematic in practical use, because of syntax and triplification errors in natural sentences. The fact that the target DB is not a well-structured ontology, but loosely linked data, is also problematic, since advance knowledge of the ontology structure is required. Thus, the following approaches that use shallow linguistic analysis are proposed for this purpose.

FREyA [10] was originally developed as a natural language interface for ontology search. In several respects it is similar to our system. For example, both FREyA and our system match

the words from a sentence with resources and properties by using a string similarity measure and synonyms from WordNet, and both improve accuracy based on user feedback. However, FREyA converts the sentence to a logical form using ontology-based constraints assuming completeness of the ontology used in the target data.

By contrast, DEQA [11] adopts an approach called Template-Based SPARQL Query Generator [12]. It takes prepared templates of SPARQL queries and converts the sentence to fill the slots in the template (not the ontology constraint). DEQA is also applicable to a specific domain (real estate search) and exhibits a certain degree of accuracy. Unlike our system, however, it does not incorporate any social approach such as the use of user feedback.

PowerAqua [13][14][15] also originated as a natural language interface for ontology search and performs a simple conversion to basic graph patterns called Query-Triples, matching of words from the sentence with resources and properties using a string similarity measure and synonyms from WordNet. When used with open data, PowerAqua also introduces heuristics according to the query context to prevent decreased throughput. It is the research most similar to our system, and addresses the issue that is called 'open-schema' in this paper, which means identifiers and properties used in the Linked Data are unknown. It then proposes mapping techniques for querying large-scale contents across multiple domains. However, improvement of the selection of mapping by user feedback is identified as a future issue [15].

In terms of commercialized systems, voice assistants such Apple's Siri and xBrainSoft's Angie have become popular recently. Both offer high-accuracy voice recognition functions and are good at typical tasks such as calling up handset capabilities and installed applications, which are easily identified from the query. These voice assistants correctly answer the question in the case that the information source is a well-structured website such as Wikipedia. Extracting the information from unstructured websites, however, often fails and they return the search engine results page (SERP), and thus the user needs to tap URLs from the list. In contrast, our system focuses on the information search using LOD as the knowledge source, and raises the accuracy using the user feedback.

In terms of mobile applications in agriculture, Fujitsu Ltd. offers a recording system that allows the user to simply register work type by buttons on a screen with photos of the cultivated plants. NEC Corp. also offers a machine-to-machine (M2M) service for visualizing sensor information and supporting gardening/farming diaries. Both systems address recording and visualization of the work, although our system is aimed at the search of cultivation knowledge on site by means of a voice-controlled QA system.

III. Problems and Approaches to Open-Schema Data

In the classification of interactive systems, our QA service is in the same category as Siri, which is a DB-search QA system. However, Siri is more precisely a combination of a closed DB and an open Web-search QA system, whereas our system is an 'open' DB-search QA system. Although the detailed architecture is described in the next section, the basic operation is to extract a triple such as subject, verb, and object from the query sentence by using morphological analysis and dependency parsing. Figure 1 shows a conversion from a dependency tree to a triple. Any query words (what, where, when, why, etc.) are then replaced with a variable and LOD DB is searched. SPARQL is based on graph pattern matching, and this method corresponds to a basic graph pattern (one triple matching). At data registration, if there is a resource corresponding to the $subject$ and a property corresponding to the $verb$ from the user statement, a triple, which has the $Object$ from the user statement as the Value, is added to the DB.

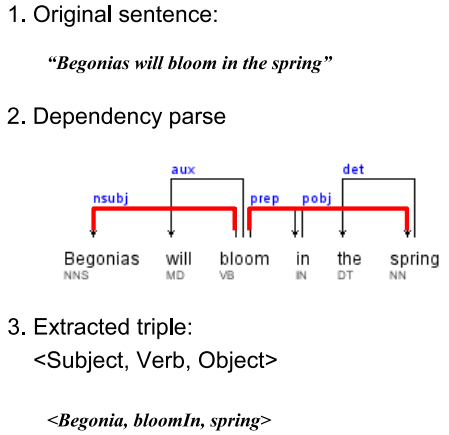


Figure 1. Conversion from dependency tree to triple.

Although DB-search QA systems without dialog control have a long history, there are at least the following two problems because the data schema is 'open'.

A. Mapping of Query Sentence to LOD Schema

The schema of the LOD, which is our knowledge source, is not regulated by any organization, and there may be several properties with the same meaning and a new property may suddenly be added. In addition, we assume searching over multiple LOD sets made by different authors. In this open-schema data, mapping between the verb in the query sentence (in Japanese) and a property in the LOD is unknown, although at least the schema is given in advance in the case that a closed DB is a knowledge source. Therefore, the score according to the mapping degree cannot be predefined.

Thus, we use a string similarity and a semantic similarity technique from the field of ontology alignment to map verbs to properties, and then attempt to improve the mapping based on user feedback. We first register a certain set of mappings {verb (in Japanese), property} as seeds in the Key-Value Store (KVS), since the KVS is faster than the LOD DB. If a verb is unregistered, we then do the following (Figure 2):

(1) Expand the verb to its synonyms using Japanese WordNet ontology, and then calculate the Longest Common Substring (LCS) with the registered verbs to use as the similarity.
(2) Translate the new verb into English, and calculate the LCS of the English with the registered properties.
(3) If we find a resource that corresponds to a subject in the query sentence in the LOD DB, we then calculate the LCSs of the translated verbs with all

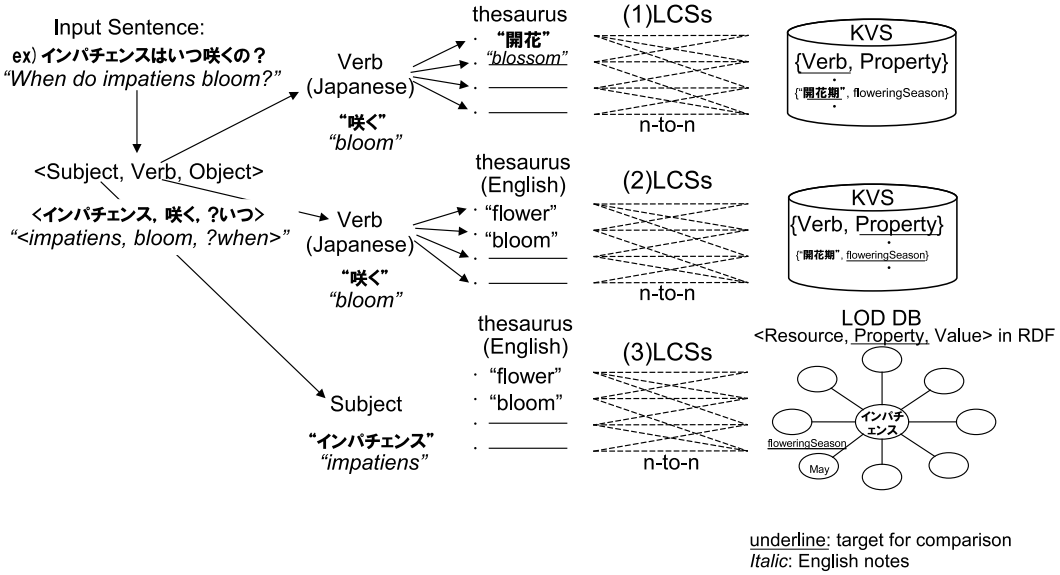


Figure 2. LCS calculation process with examples.

the properties belonging to the resource, and create a ranking of possible mappings, which has a property correspoinding to the original verb in the first position, and is arranged in descending order of confidence value (0 by default), and in descending order of the above LCS values if there are properties with the same confidence value.

(4) The user feedback, indicating which property was actually viewed, is sent to the server, and the corresponding mapping of the new verb to the property is registered in the KVS.

(5) Since the registered mappings are not necessarily correct, we add the number of pieces of feedback to the confidence value of the mapping, and recalculate the ranking of the mapping to improve the N-best accuracy (refer to Section IV-D).

B. Acquisition and Expansion of LOD

Even for an open DB, it is not easy for an ordinary user to register new triples in the DB. Therefore, we provide an easy registration method that uses the same extraction mechanism as triples from statements.

We also provide an automatic registration method of the user context information to support data registration by the user. When the user registers a triple in the DB, the sensor data are automatically aggregated by using the smartphone's built-in sensors, and the context information related to the triple is inserted in the DB after the semantic conversion of the sensor data. Although Twitter provides a function for attaching geographical information to tweets, this method is available with a greater variety of context information. By using this method, the user can register not only the direct assertion, which is an object in the user statement, but also several items of background information at once. We describe examples of the sensor data and the corresponding context information in the next section. This is an approach to collect the necessary data from side effects of the user actions (the registration in this case), and corresponds to a typical method in Human Computation mechanisms.

By contrast, we also attach the Twitter ID of the registrant to the data as the creator in order to heighten the feeling of contribution on the part of the user who shares the significance of building the 'Web of Data'. This is another method in Human Computation mechanisms. These efforts further promote the social user participatory approach.

We have also been developing a semi-automatic LOD extraction mechanism from webpages for generic and specialized information; this mechanism uses Conditional Random Fields (CRF) to extract triples from blogs and tweets. As Minh et al. [16] shows, it has achieved a certain degree of extraction accuracy.

IV. Development of Applications for Fieldwork Support

This section shows the implementation of our service and an application. The applications of QA systems include interactive voice response, guide systems for tourists and facilities, car navigation systems, and game characters. However, these all use closed DBs and would not be the best match for an open DB. In addition, since our system does not currently incorporate dialog control such as Finite-State Transducers (FSTs), problem-solving tasks such as product support are also difficult. Thus, we focus on searching for information as described in the previous sections and introduce the following applications.

1) General Information Retrieval
DBpedia [17] has already stored more than one billion triples, and there are 31 billion triples on the Web, so part of the information people browse in Wikipedia can be retrieved from LOD.

2) Recording and Searching Information for Fieldwork
Since the system allows user registration of information, the information relevant to a specific domain can be recorded and searched, including for agricultural

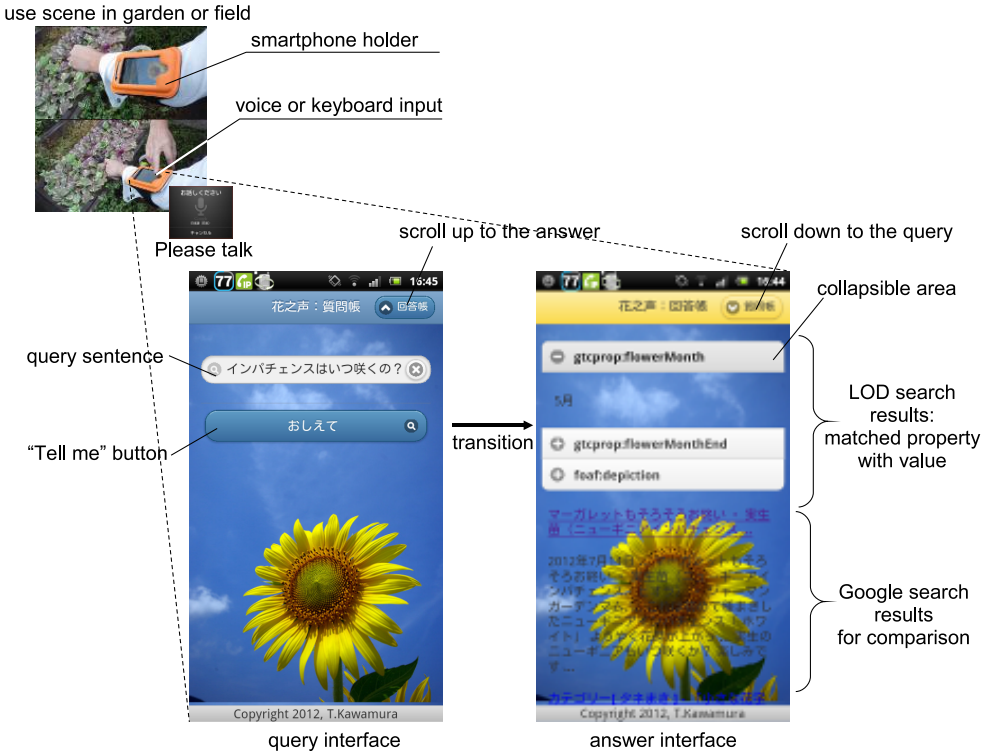


Figure 3. Service interface of Flower Voice.

and gardening work, elevator maintenance, factory inspection, camping and climbing, evacuation, and travel.

3) Information Storage and Mining Coupled with Twitter
If we focus on information sharing, it is possible that when a user tweets using a certain hash tag (#), the tweet is automatically converted to a triple and registered in the LOD DB. Similarly, when the user submits a query using a hashtag, the answer is mined from the LOD DB, which stores a large amount of past tweets. This would be useful for the recording and sharing of word-of-mouth information and life-log information.

Although the above (1) is our purpose mentioned in the introduction, we introduce an application of our QA system from the second perspective in the following section to evaluate the system in a limited domain, which is Flower Voice to answer queries on agricultural or gardening work concerning diseases and pests, fertilization, maintenance, etc.

A. Flower Voice

Urban greening and urban agriculture have been attracting attention owing to the rise of environmental consciousness and a growing interest in macrobiotics. However, the cultivation of greenery in a restricted urban space is not necessarily a simple matter. Beginners, who have no gardening expertise, have questions and encounter difficulties in several situations ranging from planting to harvesting. Although the user could employ a professional gardening advisor to solve these problems, this would involve costs and may not be readily available in urban areas. Moreover, such work cannot be fully planned and the gardener needs to respond to the current status of the plants on site, which is highly dependent on the local environment. However, searching the Internet using a smartphone is disadvantageous in that one must input keywords and iteratively tap and scroll through SERP to find the answer. Therefore, we developed Flower Voice, which is a QA service for smartphones that answers questions about agricultural and gardening work. Furthermore, we provided a mechanism for registering the work of the user, since data logging is the basis of precision farming according to the Japanese Ministry of Agriculture. This is a tool for searching information and for logging by voice using smartphones for agricultural and gardening work. Figure 3 shows a service interface of Flower Voice. It automatically classifies the speech intention (Question Type) of the user into the following four types (Answer Type is a literal, Uniform Resource Identifier (URI), or image).

1) Information Search
Search for plant information in the LOD DB. For example:
Q: “When do impatiens bloom?”
A: Flowering Season: May

2) Information Registration
Register new information for a plant that does not currently exist in the LOD DB or add information to an existing plant. For example:
Q: “Geraniums are annuals.”
A: annual: True

3) Record Registration
Register and share records of daily work. Since data logging is important for farming, it would be useful to add sensor information together with the registered

record. However, the verbs that can be registered are limited to the predefined properties in the DB (see the next section). For example:

Q: "I put fertilizer on the tulips."
A: Fertilizing Day: Oct. 12

4) Record Search
Search through records to remember previous work and view the work of other people. For example:

Q: "When did I put fertilizer on the tulips?"
A: Fertilizing Day: Oct. 12

B. Plant LOD

The LOD used by Flower Voice is called Plant LOD, and consists of more than 10,000 resources (species) under the Plant Class in DBpedia and 104 Japanese resources that we have added. We have also added 37 properties related to plant cultivation to the existing 300 properties. In terms of the LOD Schemas for registering records, we prepared properties mainly for recording dates of flowering, fertilizing, and harvesting. Figure 4 illustrates Plant LOD, which is an extension of the LOD used by Green-Thumb Camera [18], which was developed for introducing plants (greening design). Plant LOD is now stored and publicly available at Dydra.com [19], although literal values are described in Japanese.

C. System Architecture

Figure 5 shows the architecture of Flower Voice. The user can input a query sentence by Google voice recognition or keyboard. The system then accesses the Yahoo! API for Japanese morphological analysis to extract a triple using the built-in dependency parser, and generates a SPARQL query by filling in slots in a query template. A similar process also works for English sentences, although the morphological analyzer and dependency parser must be changed to, for example, Berkeley Parser [20]. The search results are received in Extensible Markup Language (XML) format. After searching the {verb, property} mappings registered in Google Big Table and accessing the Microsoft Translator API and Japanese WordNet Ontology provided by the National Institute of Information and Communications Technology (NICT), the LCS values for each mapping are calculated as described in Section III-A. The order of matching is firstly matching the $subject$ against resources by tracing 'sameAs' and 'wikiPageRedirects' links, and then searching for $verb$ matches with the properties of the resources. A list of possible answers is then created from the pairs of properties and values with the highest LCS values. The number of answers in the list is set to three because of constraints on the client UI. The results of a Google search are also shown below in the client to clarify the advantages and limitations of the QA service by comparison. User feedback is obtained by opening and closing a collapsible area in the client, which gives a detailed look at the value of the property (but only the first click). The type of feedback is classified into 'implicit' graded relevance feedback [21][22], since the user is only viewing the result without knowing that the selection will be used for the improvement of accuracy. During searches, feedback updates the confidence value of a registered mapping {verb, property} or registers a new mapping. During registration, the feedback has the role of indicating to which of three properties the $object(value)$ should be registered. The client UI displays the results. Text-to-speech has not been implemented yet.

The automatic registration method of the user context information is realized by acquisition of sensor data and semantic conversion based on the LOD Schema. The sensor data are obtained by JavaScript running on the smartphone, except for Osaifu-Keitai that is FeliCa (a specification of Near Field Communication) mobile payment. Table I shows examples of the sensor data and the corresponding context information. Note that although the clock and Osaifu-Keitai are not the sensors, these are included in the table for showing the mapping with the context information. Furthermore, Points of Interest (POIs) and Weather are obtained by accessing Yahoo! Open Local Platform and Japan Meteorological Agency based on the Global Positioning System (GPS) information. The POIs specify location names (buildings, businesses, train stations) in the vicinity of the location.

TABLE I. MAPPING OF SENSOR AND CONTEXT INFORMATION.

Sensors	Context Info. that can be obtained
Clock	Date, Time
GPS	Location, Nearby POI
(Combination of the above two)	Weather, Temperature, Humidity
Illuminance	Space{Indoor, Outdoor}
Acceleration	Status{Moving, Stop}, Walking Time&Distance

We prepared the LOD schemas (properties) corresponding to the above context information, and once the sensor information is retrieved, we convert it to the property value with the designated data types, namely, literal and integer, that are predefined by the schemas. For example, when a user registers a triple describing "a flower has blossomed", the sensor data for the location is converted to literals: one for the temperature is converted to an integer, and one for the space is translated to "Indoor" or "Outdoor". Then, the context information, such as **gtcprop:flowerAddress** (location), **gtcprop:flowerDateHighTemp** (highest temperature of the day), **gtcprop:flowerDateLowTemp** (lowest temperature of the day), or **gtcprop:flowerSpace** (space of the flower), is automatically registered in the LOD DB.

Figure 6 shows the combinations of properties intentionally registered by the user and the context information automatically obtained by the sensors. For example, when the user registers a flowering date with **gtcprop:flowerDate** property, the space and location information of the user and the weather on that day are automatically obtained. The links of the property and the context information can be easily adjusted according to the purpose of application. Flower Voice currently does not use the context information related to the user actions such as number of steps, walking distance and walking time. Therefore, there are unlinked contexts in the figure.

We have also added an advanced function for changing the LOD DB that is searched by the user input to a SPARQL endpoint as entered in an input field of the client UI, although the change is limited to searches. This is not compatible with all servers because the query is based on predefined templates and the results are received in XML format. Some servers also require attention to latency. Endpoints that have been confirmed include DBpedia Japanese [23], Data City SABAE

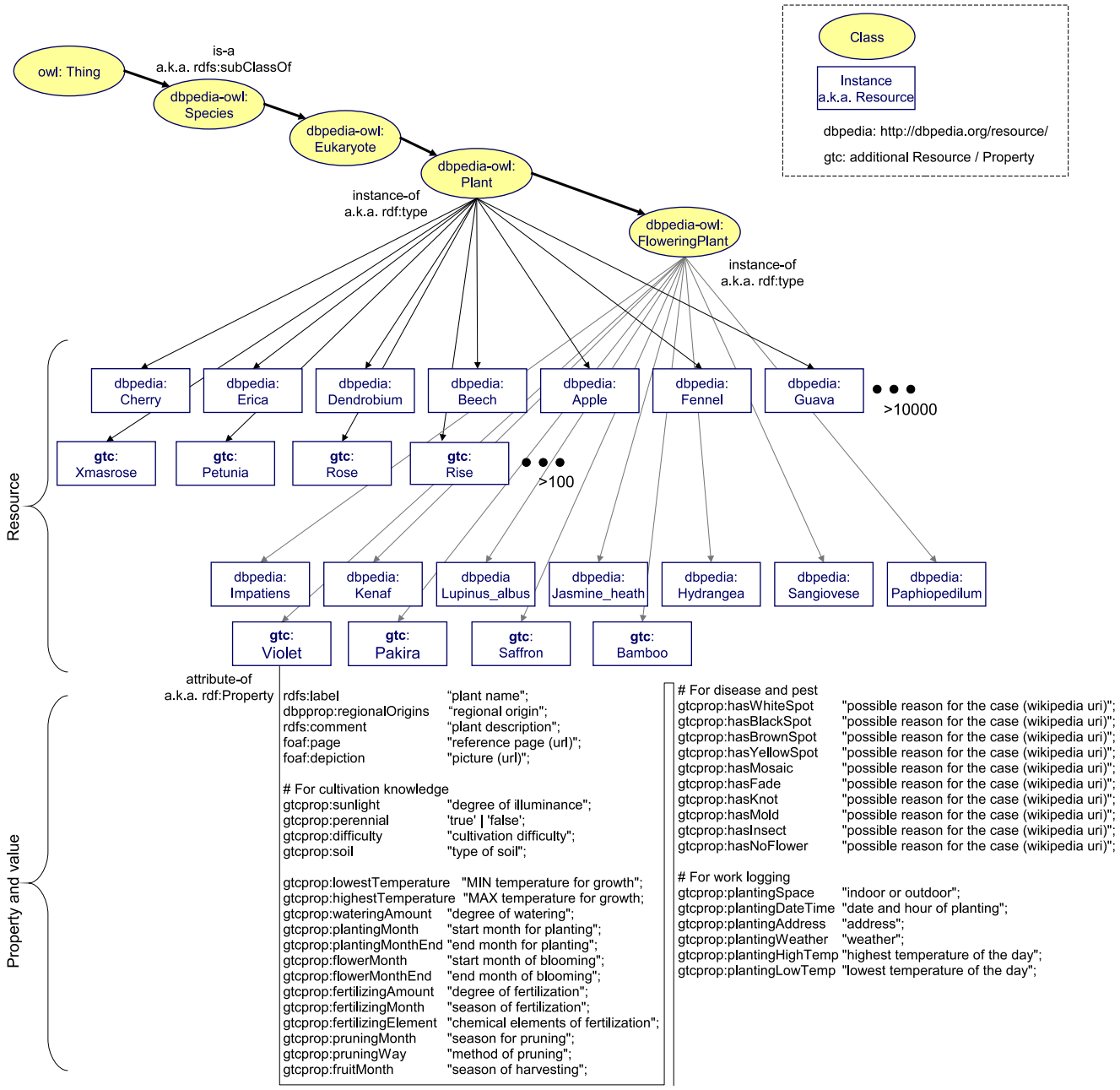


Figure 4. Part of Plant LOD, which represents examples of <Resource, Property, Value> and classes.

[24], Yokohama Art LOD [25], etc. Users can also manually register {verb, property} mappings. If the property that a user wants does not appear in the three answers, the user can input a {verb, property} mapping in an input field. The mapping is then registered in the KVS and will be searched by the next query. Although this function targets users who have some expertise for dealing with LOD, we are expecting to discover unanticipated use cases once the system is open to users.

Flower Voice is available from our website (in Japanese) [26], and almost 500 users have used it for at least one query so far (Flower Voice won a Judges' Special Award in LOD Challenge Japan 2012).

D. Evaluation of Accuracy Improvement

We conducted experiments on the current system to confirm the search accuracy, and how the accuracy is improved by the user feedback mechanism described in Section III-A. Note that if a sentence is composed of more than two triples, it must be queried as separate single sentences. The intention of the speech, such as searching or registration, is classified by the existence of question words and the use of postpositional words, not by intonation. Sentences need to be literally described regardless of whether they are affirmative or interrogative. In the experiment, we asked several experienced gardeners to select frequently asked questions from their daily work, and collected 99 query sentences (and the preferred answers). The

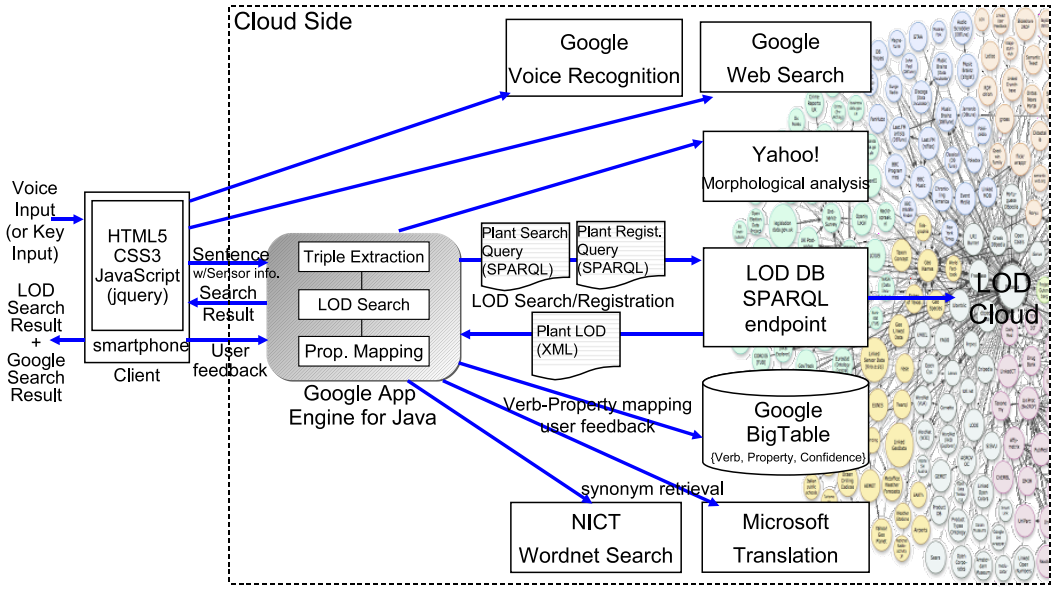


Figure 5. Mashup architecture.

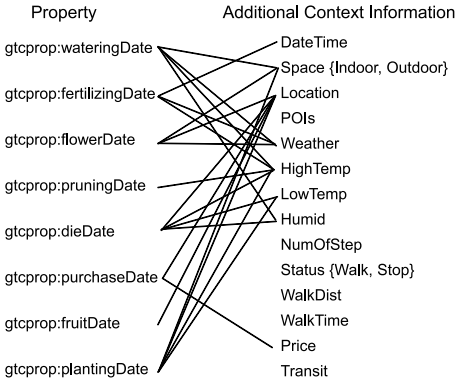


Figure 6. Registered property and additional context information.

query sentences include:

- "I brought some impatiens yesterday."
- "When did I water the lantana?"
- "Do impatiens need fertilizer?"
- "What kind of flower is impatiens?"
- "What is a good fertilizer for impatiens?"
- "Do impatiens like partial shade?"
- "Can I grow wild strawberries in the house?"
- "How much sunlight do pumilas need?"
- "Why are the leaves of my impatiens turning yellow?"
- "Do impatiens come from Africa?"
- "Do impatiens need magnesium?"

Although there were no duplicate sentences, sentences having the same meaning at the semantic level were included. We then randomly constructed 9 test sets, each consisting of 11 sentences. We first evaluate one test set randomly selected, and give the correct feedback, which means registering {verb, property} mappings and updating the confidence value for one of the three answers for each query. We then proceed to the next set. After evaluating the second test set, we clear the effects of the user feedback and repeat the above again from the first set. The difference of the accuracy between the first and the second set corresponds to the improvement by the user feedback. The results are shown in Table II. We assume that query sentences are correctly entered, since in practice Google Voice Recognition returns the possible results of the recognition, and users can select the correct sentence in a dialog, or start again from speech.

TABLE II. ACCURACY OF SEARCH.

	Failure			Success	
	no Res.	no Prop.	triplification error	1-best	3-best
1st Set (ave)	18.2%	0%	9.1%	54.5%	72.7%
2nd Set (ave)				72.7%	72.7%

In the table, "no Res." means that there was no corresponding resource (plant) in the Plant DB, and "no Prop." means no property corresponding to the verb in the query sentence. "triplification error" indicates failure to extract a triple from the query sentence in the case of a long complex question, etc. N-best accuracy is calculated by the following equation:

$$N - best\ precision = \frac{1}{|D_q|} \sum_{1 \le k \le N} r_k, \qquad (1)$$

where $|D_q|$ is the number of correct answers for query q, and r_k is an indicator function equaling 1 if the item at rank k is correct, zero otherwise. In the case of 3-best, the three answers are compared to the correct answer, and if any one of them is correct, then the result is regarded as correct.

We found that approximately 20% of the queries were for unregistered plants, and the prepared properties covered all of

the queries. The current extraction mechanism is rule-based, and approximately 10% of the queries were not analyzed correctly. Although the queries are in a controlled natural language since the queries need to be literally described as single sentences, we found that 90% of questions are allowed in our system. We are planning to extend the rules and use CRF [16] for further improvement.

The N-best accuracy can be increased by increasing the data amount, such as resources and properties in the Plant LOD and {verb, property} mappings, and so the base accuracy of the first set is not particularly important. However, by comparing the results for the first set with the second set, we can confirm that the accuracy of 18.2% was improved by the user feedback. We should note that the result that 1-best accuracy is equal to 3-best accuracy means all the correct answers are in the first position. Those are of course within the first three positions.

We expect that the number of acquired {verb, property} mappings will form a curve according to the number of trials that saturates to a domain-dependent value. In this domain, we found that an average of 0.09 new mappings was acquired per trial (query) from an initial 201 mappings in the DB. More detailed analysis will contribute to the bootstrap issue of applications in other domains.

E. Evaluation of Data Acquisition

We also conducted an experiment on the system to confirm effectiveness of the data acquisition. In the experiment, we first collected 44 sentences for registration from experienced gardeners, and then registered them in the DB. The same as in the previous experiment, we do not consider voice recognition errors. We assume that user feedbacks indicating properties for registering the context information are correctly entered. The results are shown in Table III.

TABLE III. EFFECTIVENESS OF ADDITIONAL CONTEXT.

Failure		Success		
no Prop.	triplification error	ratio	num. of additional context	num. of useful context
0%	9.1%	90.9%	9.3 triples per registration	3.4 triples per registration

In the table, “no Prop.” means no property corresponding to a verb in a query sentence. “triplification error” indicates failure to extract a triple from the query sentence. However, if there was no corresponding resource (plant) in the Plant DB during the registration, the resource is automatically created, and so “no Res.” does not happen in this experiment. Furthermore, “num. of additional context” means how many triples for the context information on average are automatically added with a triple that is successfully registered. Note that all the context information shown in Figure 6 is not necessarily obtained in practice because of the status and timing of the sensors. “num. of useful context” means the number of triples that the experienced gardeners considered useful among all the additional context information. The following are examples of useful context information.

wateringDate–Location, HighTemp, Space: By this combination, useful data to analyze correlation among watering frequency, circumstances and seasons would be collected.

flowerDate, fruitDate, dieDate–Address, Weather, HighTemp, LowTemp, Space: By these combinations, useful data regarding a process ranging from flowering and fruiting to dying, that depends on weather changes in each area would be collected.

pruningDate, flowerDate, fruitDate–Address, HighTemp, LowTemp: Correlation ranging from flowering and fruiting to pruning can be investigated based on the data.

hasWhiteSpot–Humid: Risk of infestation by red spider mites would be anticipated by drying of the planting space.

As a result, by automatically adding the context information as the side effects of the user registration, we confirmed that the useful triples have been increased 3.4 times. Describing them in RDF and sharing in a cloud DB allows people who have different viewpoints to analyze the relation of data from their own perspectives.

F. Evaluation of Computational Performance

Finally, we conducted experiments on computational performance of the system. This service is currently running on 1 CPU with 55.1 Mbytes memory of Google App Engine 1.8.4, where the 1 CPU corresponds to 1.0-1.2 GHz 2007 Opteron. Since it is difficult to compare the performance with other services, we evaluated the performance of the proposed functions, the performance of scalability, and the performance when the registered properties will increase in the future.

1) Performance by function: Table IV presents processing time of each function, which is shown in Google App Engine’s part of Figure 5. This result is the average time of eleven search queries except for the first query, since the processing time for the first query includes process instantiation, etc. Moreover, each query is about different plant species, since the plant data that are once queried are cached in the process, and then not searched again in the LOD DB. Query samples are described in Section IV-D. As a result, it needed almost 0.3 (sec) to make a SPARQL query from a natural language sentence, including an access to the morphological analyzer, and 0.8 (sec) for retrieving a plant data from the LOD DB, but once the data is loaded, it takes 0.7 (sec) for mapping properties to a verb in the sentence according to the algorithm mentioned in Section III-A.

TABLE IV. PERFORMANCE BY FUNCTION.

Function	(ms)
Triple Extraction access to Yahoo! Morphological analysis	315.4
LOD Search access to Dydra DB	771.2
Property Mapping access to Google Big Table MS Translation NICT Wordnet Search	663.6
Total	1750.2

2) Performance by simultaneous queries: To evaluate the service scalability, we measured the time and its increase for processing multiple search queries simultaneously. Figure 7 presents the processing time of 5, 10, 15, and 20 simultaneous queries after the first query. As well as the above evaluation, each query is about different plant species. As a result, we found that the processing times are almost unchanged regardless of the number of queries. But the queuing time of

a server process to handle the query linearly increases, and thus the total time increases linearly too. However, the service is currently running on a CPU, and processes for handling multiple queries are increasing in parallel. Therefore, we can manage this issue by increasing the number of CPUs since a rate of increase is in a linear fashion. Moreover, in the experiment the setting of queries for different plants is heavier than in actual use. In practice, many queries are about plants that have already been searched and cached. In such cases, the time for LOD Search will become 0 – 1 (ms).

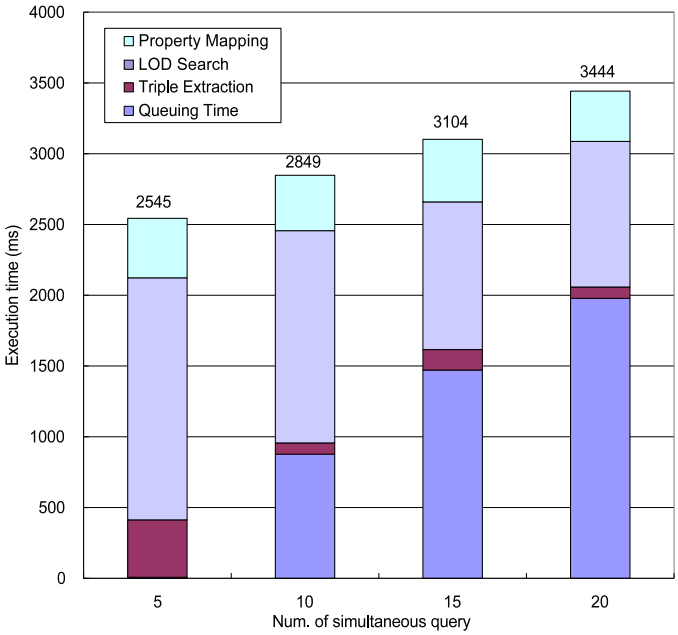


Figure 7. Performance by simultaneous queries.

3) Performance by property types: Furthermore, when the types of properties increase in the future, the calculation cost of mapping the properties to a verb will also increase accordingly. As shown in Table IV, the time to make an ordered list of the properties corresponding to a verb based on the LCS values is currently less than 0.7 (sec). In addition, the number of calculations of an LCS value and the number of the value comparisons between two properties will increase by increment of a property. According to the algorithm mentioned in Section III-A, however, the calculation cost of both processes will simply increase, and is assumed to be $O(N)$. Therefore, we can sufficiently deal with this issue by increasing the number of CPUs on a cloud platform in practice, as well as by the above evaluation. Moreover, the property generally means the type of information, and then substantial increase of the property type is not expected. On the contrary, the diversity of the properties will contribute to improved accuracy in the search after all.

V. Conclusion and Future Work

This paper proposed a query answering system, which accepts Japanese query sentences, translates them to RDF triples and sends SPARQL queries to LOD as a knowledge source. We then developed and evaluated an application implemented on a mobile phone in the context of gardening advice for people. It also features a social approach, namely, the improvement of accuracy based on user feedback and the acquisition of new data from user context information. In experiments, we confirmed that the search accuracy of 18.2% was improved by the user feedback, and the useful triples have been increased 3.4 times by the context information. Finally, we also evaluated the scalability of the service. Note that since comparison with other approaches under the same condition is problematic in the above experiments, please refer to related work section for comparison.

The proposed system is related to a number of works described in Section II. However, we assumed that the data is open, and so the schema is not given in advance. Thus, we adopted shallow linguistic analysis with the aim of achieving data portability and schema independence. In comparison with other research on shallow linguistic analysis, the novelty of our system is use of a social approach. In particular, it corresponds to improvement of search accuracy by the use of user feedback in the 'open-schema' scenario, which is described in Section III-A. Therefore, we currently do not rely on ontology-based mapping techniques such as [27]. This, however, is both a strength and a limitation of our system. The ontologies behind the LOD vary, and some of them are not structured well, and thus query expansion is not necessarily successful. In contrast, the use of user feedback achieved a significant improvement of the accuracy after the repetition of queries. This is the most important point, and the reason that our application attracts many accesses. That is, it does not fail twice. However, since the proper adaptation of the ontology will raise the accuracy of the first query, we will address this issue in the future.

As a lesson learnt from the application development, we should consider use of the context information for information and record search. We implemented a method for registering user context information, which is converted from sensor data in order to acquire new data. However, it will be possible to use the context information for refining the search results to fit the user's current environment during the search.

Since we made this service available without registration, user satisfaction has not been investigated. However, almost all users' responses to this service have been favorable. Some users commented that it will no longer be necessary to check gardening/farming diaries, since every task is recorded by speech and sensors, and can be queried in the garden. In addition, an opinion was expressed that voice control is suitable for this work, since users typically have dirty hands and do no need to be shy because there is usually no one watching or listening in the vicinity. In the future, we will collect users' opinions on this application, and apply the system to domains other than agriculture.

References

[1] T. Kawamura and A. Ohsuga, "Query answering using user feedback and context gathering for web of data," Proc. of 3rd International Conference on Advanced Communications and Computation (INFOCOMP 2013), pp. 79-86, Nov. 2013.

[2] B. Ell, D. Vrandecic, and E. Simperl, "SPARTIQULATION: verbalizing SPARQL queries," Proc. of Interacting with Linked Data (ILD), pp. 50-60, May 2012.

[3] A. Simitsis and Y. E. Ioannidis, "DBMSs should talk back too," Proc. of 4th biennial Conference on Innovative Data Systems Research (CIDR), Jan. 2009.

[4] P. Haase, D. Herzig, M. Musen, and D. T. Tran, “Semantic wiki search,” Proc. of 6th European Semantic Web Conference (ESWC), pp. 445-460, May 2009.

[5] S. Shekarpour et al., “Keyword-driven SPARQL query generation leveraging background knowledge,” Proc. of International Conference on Web Intelligence (WI), pp. 203-210, Aug. 2011.

[6] D. T. Tran, H. Wang, and P. Haase, “Hermes: data web search on a pay-as-you-go integration infrastructure,” J. of Web Semantics Vol.7, No.3, pp. 189-203, Sep. 2009.

[7] P. Cimiano, “ORAKEL: a natural language interface to an F-logic knowledge base,” Proc. of 9th International Conference on Applications of Natural Language to Information Systems (NLDB), pp. 401-406, Jun. 2004.

[8] P. Cimiano, P. Haase, J. Heizmann, and M. Mantel, “Orakel: a portable natural language interface to knowledge bases,” Technical Report, University of Karlsruhe, pp. 1-77, Mar. 2007.

[9] M. Wendt, M. Gerlach, and H. Duewiger, “Linguistic modeling of linked open data for question answering,” Proc. of Interacting with Linked Data (ILD), pp. 75-86, May 2012.

[10] D. Damljanovic, M. Agatonovic, and H. Cunningham, “FREyA: an interactive way of querying linked data using natural language,” Proc. of 1st Workshop on Question Answering over Linked Data (QALD-1), pp. 125-138, May 2011.

[11] J. Lehmann et al., “DEQA: deep web extraction for question answering,” Proc. of 11th International Semantic Web Conference (ISWC), pp. 131-147, Nov. 2012.

[12] C. Unger et al., “Template-based question answering over RDF data,” Proc. of 21st International Conference on World Wide Web Conference (WWW), pp. 639-648, Apr. 2012.

[13] V. Lopez, E. Motta, and V. Uren, “PowerAqua: fishing the semantic web,” Proc. of 3rd European Semantic Web Conference (ESWC), pp. 393-410, May 2006.

[14] V. Lopez, M. Sabou, V. Uren, and E. Motta, “Cross-ontology question answering on the semantic web - an initial evaluation,” Proc. of 5th International Conference on Knowledge Capture (K-CAP), pp. 17-24, Sep. 2009.

[15] V. Lopez et al., “Scaling up question-answering to linked data,” Proc. of 17th International Conference on Knowledge engineering and management by the masses (EKAW), pp. 193-210, Oct. 2010.

[16] T. M. Nguyen, T. Kawamura, Y. Tahara, and A. Ohsuga, “Building a timeline network for evacuation in earthquake disaster,” Proc. of AAAI 2012 Workshop on Semantic Cities, pp. 15-20, July 2012.

[17] DBpedia, http://dbpedia.org[accessed: 2014-11-19].

[18] T. Kawamura and A. Ohsuga, “Toward an ecosystem of LOD in the field: LOD content generation and its consuming service,” Proc. of 11th International Semantic Web Conference (ISWC), pp. 98-113, Nov. 2012.

[19] Dydra.com, http://dydra.com/takahiro-kawamura/fv[accessed: 2014-11-19].

[20] Berkeley Parser, https://code.google.com/p/berkeleyparser[accessed: 2014-11-19].

[21] T. Joachims, D. Freitag, and T. Mitchell, “WebWatcher: a tour guide for the world wide web,” Proc. of 15th International Joint Conference on Artificial Intelligence (IJCAI), pp. 770-777, Aug. 1997.

[22] Surf Canyon, http://surfcanyon.com[accessed: 2014-11-19].

[23] DBpedia Japanese, http://ja.dbpedia.org[accessed: 2014-11-19].

[24] Data City SABAE, http://lod.ac/sabae/sparql[accessed: 2014-11-19].

[25] Yokohama ART Search, http://archive.yafjp.org/test/inspection.php[accessed: 2014-11-19].

[26] Flower Voice (in Japanese), http://www.ohsuga.is.uec.ac.jp/~kawamura/fv.html[accessed: 2014-11-19].

[27] V. Lopez, V. Uren, M. Sabou, and E. Motta, “Is question answering fit for the semantic web? a survey,” Semantic Web J., Vol. 2, No. 2, pp. 125-155, July 2011.

Agile-User Experience Design: Does the Involvement of Usability Experts Improve the Software Quality?

State of the Art and a First Experiment

Lou Schwartz
Luxembourg Institute of Science and Technology
5, avenue des Hauts-Fourneaux
L-4362 Esch/Alzette, Luxembourg
e-mail: lou.schwartz@list.lu

***Abstract*—In the past decade, numerous experiments and research proposed to take the advantages of Agile and User Centred Design methods in a mixed method called Agile-User Experience Design or Agile-UX. This combination raises a number of questions. Notably, it remains unclear who should be responsible of the usability in an Agile-UX project development. After a review of the literature on Agile, User Centred Design and Agile-UX, this paper focuses on the involvement of usability experts in Agile-UX. The literature discusses the involvement of usability experts in terms of processes and work methods, but never in terms of the necessity to involve usability experts to improve the software quality. To start answering this question, an experiment was conducted to explore the necessity to involve usability experts in the team. The results are that the involvement of a User Centred Design expert improves the quality of the developed product and the users' satisfaction in Agile-UX.**

Keywords-Agile; Agile-UX; User-Centred Design; Team composition; Involvement.

I. INTRODUCTION

Agile-User Experience Design (Agile-UX) is a project management principle for software development. It is based on Agile's values and principles, and on the User-Centred Design (UCD) method. Nowadays, no official definition of Agile-UX exists, but many experiments demonstrate its value [1][4][9][13][21][25][26][27][30][31][33][34].

Many questions still arise by this reconciliation of Agile and UCD. The one that this paper will deal with in depth is the necessity to involve a usability expert in the team. In the literature, Agile-UX is implemented with the involvement of usability experts in the Agile process and with the use of methods from UCD. But, in Agile, the intervention of experts is not encouraged [21] ("*UCD provides specialized skills in U[ser]I[nteraction] design but Agile approaches prefer generalists and discourage extensive upfront design work.*"). Rather, a dissemination of skills is preferred - by means of a "*generalizing specialist*" approach - to the intervention of experts [3]. This means it is preferable that team members can do all tasks to ensure a dissemination of the knowledge, including code knowledge, in the team and no one is left without work. Generalizing experts are multidisciplinary people able to work on different aspects or technics used in the project, like development and usability [3]. Furthermore, state of the art neither justifies nor discusses the involvement of usability experts in Agile-UX in term of necessity to improve the quality of the delivered software. In this paper, the involvement of usability experts in Agile-UX is discussed by testing both approaches within two experiments: the first one fully respects the principles of Agile project management: developers should be able to manage UCD themselves, and to conduct the related methods without the intervention of a usability expert; the second option integrates a usability expert in the project team to ensure both a better UCD implementation and results. We test three hypotheses: H1: without usability expert, if the project team has awareness and some knowledge in HCI, Agile-UX gives a correct quality level about the product's usability; H2: with usability expert involved in the project team, usability of the produced product is better than in H1; H3: the dynamic of the project team is better when a usability expert is involved.

After a reminder in Section II on the background composed of the definitions of the Agile method and the UCD method and their reconciliation in Agile-UX, a focus is placed on the literature review, with the particular research question on the involvement of usability experts in an Agile-UX development process in Section III. Afterwards, the paper presents an experiment in order to check our hypotheses in Section IV.

This paper is an extension of our contribution presented at ICSEA 2013 [1]; state of the art is extended and experiment' definition and results presentation are completed.

II. BACKGROUND

To better understand our question on usability expert involvement in Agile-UX, this paper first goes back on the Agile and UCD methods and on the issues and interests to reconcile them in a mixed method called Agile-UX.

A. Agile methods

The Agile methods are management methods for software development, which are based on an iterative development of software in order to better answer to changing requirements. According to Lindvall [20], Agile

methods can be defined as iterative, incremental, self-organizing and emergent.

1) Values: The Agile methods aim at enhancing the value of the delivered product to satisfy the customer's requirements. The production is organized in iterations (or sprints) from two to eight weeks. Agile methods plebiscite the following four values defined in the Agile Manifesto [2]:

- Individuals and interactions over processes and tools.
- Working software over comprehensive documentation.
- Customer collaboration over contract negotiation.
- Responding to change over following a plan.

The Agile movement was instigated and pioneered by software developers in reaction to a frustrating history of projects being delayed, going over budget, collapsing under their own weight and stressful jobs. For the Agile manifesto founders, these problems have their origin in the excessive analysis, specifications and designs done before code writing that enabled unstable or not useful requirements and incompleteness. With the Agile methods, customers would obtain faster working software that better corresponds to their actual needs, thanks to the flexibility provided with the development process [4].

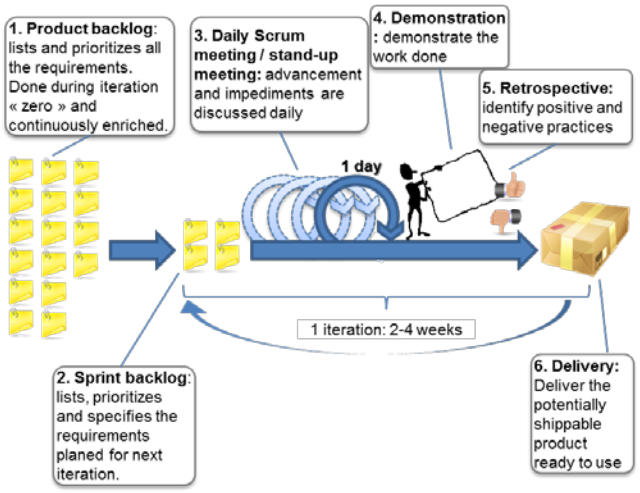


Figure 1. Scrum process.

2) Most used methods: Today, the two most used Agile methods [12] are Scrum [28] (see Figure 1) and eXtreme Programming (XP) [6], or a mix of them, including the proposed integration of Agile methods and UCD [16]. Scrum focuses on management practices instead of development or software engineering practices [19]; it is then easier adaptable for the integration of other experts' practices like UCD. This certainly explains why it is the most used in reconciliation between Agile development methods and UCD.

3) Weaknesses of the method: some Agile methods are more focused on the developers' work and on the development quality, like XP. And even if the aim of Agile methods is to satisfy the product owner, they define neither method nor good practice to achieve this objective, particularly for the needs elicitation or the design part. The needs elicitation is done by the product owner, based on his proper knowledge of the domain or of the work done by users. He can use the methods he wants, including involvement of users (e.g., by interviews, context inquiries, etc.). After that, the needs are discussed within the team to refine and prioritize them, based on the business value but also on their technical complexity or on the necessity that previous work was done to realize them.

Concerning the UI design, it depends on the openness to the usefulness and usability of developers, the customer and the consulted users, so there is no guarantee about ergonomics [7]. Indeed, the product owner and developers are often not trained on the UCD approach and the associated methods. Developers are more focused on the client's needs than on the users' needs, and a lot of time and work are required from the product owner to gather all the user's needs and feedback. Unfortunately, the product owner is not only within that function, but often he continues to work on his normal tasks as employee of the organization. Thus, product owners do not have enough time for this additional task. It is the same for developers; they often have plenty of other tasks with only a limited timeframe left to set up a user centred approach. Moreover, Agile pushes developers, and often also the product owners, to focus only on a single set of functionalities (the user stories developed during the current iteration), so they sometimes lose the holistic view, which as a consequence, presents homogeneity problems. That is why it seems a good option to involve UCD experts to ensure staying in line with the real end-users needs, to organize the UCD approach, to implement the required UCD methods and to maintain a holistic view of the final software design.

The use of the UCD principles and methods is one way to ensure answering to users' needs. Based on these assessments, it seems that Agile teams can benefit from integration of UCD methods with Agile, in particular to improve the needs elicitation and the design part.

B. *User-Centred Design*

UCD focuses on producing usable software that not only satisfies real users' needs, but also those of customers. This method, described by the ISO 9241-210 standard [18] (see Figure 2), defines the process to follow in order to produce software that meets the users' requirements. It includes notably the design and the validation phases. By nature UCD is not focused on the developers' work.

1) Principles: The principles of the UCD are listed below [18]:

- The design is based upon an explicit understanding of users, tasks and environments.
- Users are involved throughout design and development.

- The design is driven and refined by user-centred evaluation.
- The process is iterative.
- The design addresses the whole user experience.
- The design team includes multidisciplinary skills and perspectives

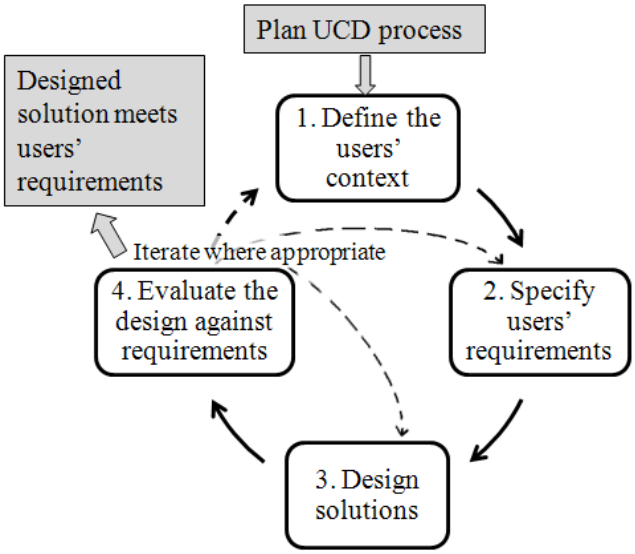


Figure 2. UCD process as described by the ISO 9241-210 standard [18].

2) UCD methods: The implementation of the UCD process involves many methods (like prototyping, observations, interviews, users' tests, etc., see ISO/TR 16982:2002 [17] for descriptions of some of them) to support, amongst other things, the users-needs' definition and the validation of the delivered software by end-users. These methods are conducted by usability experts. They select the more appropriate methods concerning the context of the project (including constraints like budget and planning, the access to users, the available skills in the teams, etc.).

Agile and UCD processes are quality processes, which have the objective to provide the most suitable software with minimal issues. They are also both iterative. Then, they seem compatible and could enrich each other. In the next section, their compatibility will be discussed.

C. *Reconciliation of Agile and UCD and research questions that arise*

Even if some Agile concerns could prevent a UCD attitude [7] (focus is often more on programming techniques and programmers, automated tests, very short iterations, fast increments and executable software as a measure), a reconciliation of both approaches is possible and has often been implemented. Since a decade, several works propose to reconcile Agile and UCD [4][7][10][19][21][22][23][29][33]. Several experiments indicate that an integration of Agile and UCD produces some interesting results [9][13][16][21][31]. As Nelson presents them, "*[XP (or Agile methods) and interaction design (or UCD) are] process[es] with similar goals but different methodology*. [23]" In fact, the two methods have a lot of compatibilities, but some impediments require adapting both to be efficient (see Table I for a synthetic view of conflicts and compatibilities between Agile and UCD). This reconciliation raises a lot of research questions. Some of them are listed in the following parts and, in the following section, the focus is placed on one particular question raised about the necessity of the involvement of UCD experts in Agile-UX.

1) Impediments to a mixed method: We can particularly note the following impediments and resulting questions.

In Agile methods, the intervention of experts is not encouraged and generalists or generalizing specialists are preferred [3][21]. UCD proposes the involvement of usability experts. So our questions are:

- Who should be in charge of UCD in an Agile project: team members or involved usability experts?
- How can usability experts be involved in the team?

In Agile, teams include a product owner, who is the customer and, de facto, the user representative [26]. In UCD this role is taken by a usability expert [21]. Our questions about this are:

- What are the responsibilities and activities of each role?
- It is necessary to keep these two roles?

Agile discourages extensive upfront design work while it is common that in UCD a deep analysis is done upfront [21]. So the questions are:

- Is it possible to reduce the first analysis done in UCD to fit the iteration duration and thus, realize this analysis during the iteration called *zero* [29]?
- Could the analysis be disseminated throughout the project?
- The deep analysis done upfront in UCD has the objective to provide a global vision and enable more homogeneity. How can a global vision and homogeneity be ensured?

UCD recommends the use of some design artefacts to facilitate communication with the project team, while Agile advocates focusing more on software developed than to produce unused documentation [9]. (Agile principle 2: Working software over comprehensive documentation [1]).

- Are some artefacts of UCD useful and simple enough to produce and understand, to improve communication without intruding upon the effort to produce working software?

The evaluation is done on different levels: from low-fidelity designs to software [26] in UCD with often only few users, and in Agile on production-ready application by real users in their real context [21].

- Are both levels of evaluation necessary?
- How to synchronize them?
- What are their specific objectives?

TABLE I. SYNTHETISIS OF CONFLICTS AND COMPATIBILITES IN AGILE AND UCD.

Agile	UCD	Compatible	Practices proposed in Agile-UX to ensure compatibility
Prefers generalists with some expert knowledge (generalizing specialists) Small team	Involvement of any kind of experts necessary for the project included UCD expert Multidisciplinary team	No	No involvement of UCD expert, but have someone with UCD knowledge in the team (developer, coach or product owner). Or - Involve when needed 1 UCD expert. Or - Involve 1 or several UCD experts throughout the project.
Product owner is customer and de facto user representative	UCD expert is user representative and de facto to a certain extent customer representative	No	UCD and product owner are necessarily involved in the project. Or - UCD expert is the product owner.
Agile discourages extensive upfront design work	In UCD often a deep analysis is done upfront	No	Reduce the upfront analysis, use the iteration *zero* to do it. Design the global vision (overall layout, navigation and look&feel) upfront, then detailed throughout the project when it is necessary and maintain the global vision. Or - Do a deep upfront analysis phase before an Agile development phase.
Value 2: Working software over comprehensive documentation	UCD recommends the use of some design artifacts to facilitate communication with the project team	No	Use only high value artifacts. And - Simplify the methods of artifacts' production. And - Simplify artifacts presentation. And - Produce the artifacts only when they are needed And - Disseminate their results at the end of each iteration Provide a visionary prototype of the final product maintained throughout the process
Evaluation is done on a production-ready application by real users in their real context	Evaluation is done on different levels: from low-fidelity designs to software, often with only few representative users	No	Both are complementary and needed
Agile is focused on code production and work of developers	UCD focuses on user interfaces and interactions and work of usability experts	No	Agile-UX proposes a reconciliation of these two points of view. Different processes are proposed to support the parallel work of UCD experts and developers
Focus on quality of the product. With regard to the Agile method used (XP, Scrum, Lean, etc.) and team skills, in Agile, the focus is placed either on the quality of the code or the quality of the product	Focus on usability and utility of the end product as measure of the quality of the product	Partially	Agile-UX requires taking into consideration factors such as utility and usability to ensure that the focus is well done on product quality
Satisfaying the customer	Concentrating on the user needs	No	Have a real end-users representative as product owner, support the product owner in the end-user needs identification and understanding thanks to the constant involvement of end-users throughout the development asked by UCD methods.
Iterative	Iterative	Yes	
Allow involvement of end-users and give access as soon as possible to working software to real end-user in their real context of user	Focus is done on end-users Feedback of end-users are essential	Yes	Can bring along more contextual and complete information to UCD experts as users' tests in laboratory or with all-comers users
Multi-disciplinary is not rejected by Agile, even if the involvement of experts has to be limited	Multi-disciplinary is a key value of UCD	Yes	Involvement of UCD experts in the Agile team when usability of the product is defined as a very important quality needed
Provides solid foundation for a user-centred attitude	User-centred attitude is a key value of UCD	Yes	
Continuous testing throughout the project of the developed software	The design is driven and refined by user-centred evaluations, and this is a key value of UCD	Yes	
Value 4: Responding to change over following a plan	Accept change: coming from users' feedback or from customers (context)	Yes	
Quality process of the code produced (bug free) and to ensure to answer the needs defined by the product owner	Quality process of the interfaces and interactions (usability) and ensuring to answer the real end-users' needs	Yes	Support the product owner on the definition of the users stories to ensure the representation of the real end-users' needs
Reduce costs of development by developping a bug free software and by the development of only the expressed needs (no more)	Reduce the cost by avoiding design errors that will reduce training time of users, avoid the rejection of software by users and decrease the risk of improving the developed software thereafter	Yes	

Agile is focused on valuable software production by ensuring the quality of the product. With regard to the Agile method used (XP, Scrum, Lean, etc.) and team skills, in Agile, the focus can be placed either on the quality of the code or the quality of the product. Regarding UCD, this focuses on user interfaces and interactions quality and work of usability experts [13][26]:

- Should the priority be given to the best practices and values of one (Agile or UCD) or both?
- What are the relations between developers and usability experts in Agile-UX?
- Do the usability experts and the developers find their place and feel well in Agile-UX?
- How to organize the development and the design work, what are the processes? Agile focuses on satisfying the customer, who is supposed to be a representative of the end-users and knows their needs. UCD focuses more on answering the user needs, while taking into account the overall context provided by the client organisation and their representatives [26].

2) Compatibilities that encourage a mixed method: Agile and UCD also have compatibilities.

Agile methods and UCD are both iterative processes even if the lengths of their cycles are different (some weeks in Agile, some months in UCD [21]).

- Can UCD and Agile cycles be synchronized?
- What are the different steps of each process and are they aligned? Agile methods allow an involvement of end-users and provide access to the working software the real end-users in their real context of use as soon as possible, which can bring along more contextual and complete information than users' tests in laboratory or with all-comers users. User feedback is also important in Agile development methods [19].
- How and when to involve users in the design and validation of the software?
- How to deal with the users' feedback?

Multi-disciplinary is not rejected by Agile, even if the involvement of experts has to be limited [21]. Indeed, as Blomkvist exposes, an Agile project culture provides a solid foundation for a user-centred attitude: *"a focus on people, communication, customer collaboration, adaptive processes and customer/user needs. [7]"*

- Continuous testing throughout the project is a good practice for both:
- How to integrate UCD tests with the Agile process and practices?
- What role do the UCD tests play in acceptance tests?
- Both accept change: coming from users' feedback or from customers [23].
- Finally, both are quality processes [14]: UCD aims to deliver quality software adapted to users, to their needs, to their context and to their tasks. Agile aims to deliver quality software without bugs and, which is adapted to the needs and constraints expressed by the product owner.
- Both have also the objective to reduce development costs: UCD by avoiding design errors, which will reduce training time of users, avoiding rejection of software by users and decreasing the risk of having to improve the developed software afterwards; and Agile by doing only what is asked and reducing the bug fixing after release [23].

3) Conclusion: To conclude, Agile methods do not cover all UCD's principles, but there is no blocking contradiction between Agile and UCD approaches and conversely. This certainly explains the increasing number of experiments or propositions of mixed methods.

In the next part of this paper, we will focus on a particular question raised in the literature review: the involvement or not of usability experts in the Agile-UX team. The next sections will deal with this question deeply through a more focused state of the art and the proposition of experiments to test the validity of our hypotheses.

Agile methods and UCD are both iterative processes even if the lengths of their cycles are different (some weeks in Agile, some months in UCD [21]).

III. USABILITY EXPERTS INVOLVEMENT IN AGILE-UX

An expert is defined in the Oxford dictionary as "A person who is very knowledgeable about or skilful in a particular area." Specialists or experts are professionals that have deep knowledge and skills concerning a particular domain, technology or methodology. They are focused on their subject of expertise.

Generalizing specialists are experts on several subjects. "They have one or more technical specialities but also a general knowledge in other areas of software development." Agile fosters this overspecialisation, by encouraging team workers to work on tasks outside their speciality [3].

UCD fosters an expert approach, while Agile rather advocates a generalizing specialist approach. How have these different approaches been mixed together in Agile UX? Through numerous experiments of Agile-UX, the question of *who is in charge of UCD* often comes up [4][9][13][15][21] [23][25][30][31][33][34]. Different options are exposed, which can be grouped as explained below.

A. Specialist approach

One or more specialists (UCD experts) are involved in the Agile-UX process. The consistency of the interventions can be different from one project to another: from some punctual interventions (often at the beginning of the project, for conducting users' tests or on demand) to a constant presence throughout the project (often following Sy's parallel tracks process).

1) One usability expert: Only a few experiments advocate the integration of only one person in charge of UCD in Agile-UX (project 1 and project 3 in [13], P1, P5, P9 and P10 in [15], project PV in [21], [30][31]).

2) A parallel team of several usability experts: In most cases, a parallel team of several usability experts is dedicated to the project ([4][9], project 2 and project 4 in [13],[23][25][33], P2 and P4 in [15], [34]). Still, they

organise the exchanges and the work between developers and designers differently.

B. Generalizing specialist approach

In a generalizing specialist approach, the product owner (project 1 in [13]) or some developers (project 3 in [13], P3, P6 and P8 in [15]) conduct also the UCD.

1) UCD expert as product owner: With regard to the UCD expert's and product owner's responsibilities, it is sometimes preferable to merge both roles (project 1 in [13], project TB in [21], defended by Beck in [23], [31][33]). The product owner, de facto, has a lot of responsibilities that can be taken into charge by UCD experts and UCD methods (see Table II) taking into consideration the UCD experts' role and responsibilities.

Furthermore, some observations show that the product owner is often overcharged with marketing and sales concerns. He often does not have the skills to manage a user-centred design, and, as a consequence, he may lose focus on a user experience vision [31].

TABLE II. HOW UCD EXPERTS CAN TAKE IN CHARGE PRODUCT OWNER RESPONSIBILITIES.

Product owner responsibilities [31][32]	**How UCD experts can take into charge product owner responsibilities**
Define the features of the product and decide on release date and content	UCD experts can define the user stories to develop based on gathered data on users, context and tasks [31].
Be responsible for the value of the product	By context studies, exchanges with the organisation on the needs and the attempted value, and observations of users and their feedback, UCD experts can define the value of the product and define priorities.
Prioritize features according to market value	
Can change features and priorities every 30 days	UCD expert accepts changes and modifies designs when it is necessary, based on users feedback. He can modify user stories and prioritizations according to new analysis.
Accept or reject work results	UCD experts use users' tests, expert validations to reject or accept the work results. These methods can be part of the acceptance tests.
Negotiate with all stakeholders	
Communicate with users and train them	This is also a responsibility of the UCD expert role.

TABLE III. WHO IS IN CHARGE OF UCD IN AGILE-UX, SYNTHETISIS OF THE LITERATURE REVIEW.

		Who is in charge of UCD in Agile-UX team?				**Consistency of the intervention**		**Good practice**
		Specialist approach		***Generalizing specialist approach***				
		1 usability expert	*Parallel team of UCD experts*	*1 UCD expert as product owner*	*UCD is done by developer(s) or other team member(s)*	***Punctual intervention***	***Constant presence throughout the project***	***Pair designing***
Armitage [4]			X					
Chamber-lain [9]	Project I		X					
	Project M		X					
	Project S		X					
Ferreira [13]	Project 1			X				
	Project 2		X					
	Project 3				X	X		
	Project 4		X					
McInerney [21]	[21]				X		X	
	Project MG					X		
	Project PV				X	X		
	Project TB			X	X			
Nelson [23]			X	X				
Nodder [25]			X					
Nummiaho [26]							X	X
Singh [31]				X				
Sy [33]			X	X			X	
Fox [15]	P1	X						
	P2		X		X			
	P3				X			
	P4		X		X			
	P5	X						
	P6				X			
	P7							
	P8				X			
	P9	X						
	P10	X						
Patton [27]					X			
Silva [30]		X					X	
Wale-Kolade [34]			X		X	X	X	

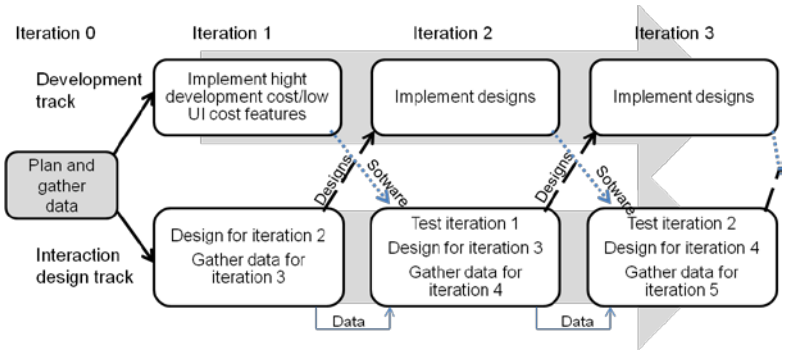


Figure 3. Parallel tracks of work of development and interaction design proposed by Sy [33].

2) Team members as responsible of the UCD process: The last possibility is to make some team members responsible for the UCD process. It is also the more close to the Agile visions: have team members who are generalizing specialists, who thus combine, for instance, skills in development and in UCD (project 3 in [13], P2, P3, P4, P6 and P8 in [15], [21][27][34]).

C. Work organisation

In addition to the question on the distribution of UCD responsibilities, the organization of UCD tasks is addressed in the literature.

Sy [33] proposed a parallel track organisation of work: designers work with one or two iterations ahead of developers (see Figure 3). To implement this proposition, several usability experts are needed, because of the amount of work. Several teams adopt this work organisation ([4][9], project 3 and 4 in [13], [26][34]), but sometimes with only one UCD expert [30].

In Agile methods, it is possible to dedicate a spike (an iteration to focus on a particular problem like testing a new technology) to usability exploration. Still, it is not a good solution to maintain a constant pace [25].

Some projects also occasionally involve UCD experts on some particular points (projects MG & PV in [21], [34]); this is close to an organisation by spikes. But, for McInerney [21], it is important that the usability expert is available *on call* at all times, which may be impossible if the usability expert works on several projects simultaneously.

Some other projects integrate usability in the iteration without real planning (see [P3.290] in [13]).

D. Synthesis

In the literature, both modalities can be founded: involving or not a UCD expert. We can note a preference for the involvement of UCD expert(s).

To summarize, in literature, Agile-UX teams involve at least one UCD expert most of the time (see Table III). His role, the consistency of his intervention, and the synchronization of his intervention with the developers' work are not fixed, even if the parallel tracks of design and development seem to be the most adopted practice. In the studied papers, there is no mention of why a UCD expert in Agile-UX should be involved. There is no reference on the fact that it could or not be better to involve a UCD expert in the team. This can raise the following question: is it necessary to involve usability experts in the team, or is involving team members with some knowledge on usability sufficient? This is what we tested in the implemented experiment presented in the following section.

IV. EXPERIMENT

After the literature review and interviews with Agile professionals, we focused on the question of the usability expert involvement in the team. Literature contains a lot of experiments on the involvement of usability experts in an Agile team, but they are more focused on the process and methods used than on the necessity to involve usability experts in Agile-UX. Based on this observation, we propose the following hypotheses to check the importance to involve usability experts in Agile-UX team in order to improve the produced software's quality in terms of usability:

H1: without usability expert, and if the project team has awareness and some knowledge in HCI, Agile-UX renders a correct quality level about the product's usability.

H2: with usability expert involved in the project team, the usability of the produced product is better than in H1.

H3: the dynamic of the project team is better when a usability expert is involved.

We retrospectively and qualitatively question these hypotheses through an experiment. The focus is done only on the usability of the final product, laying aside any potential overhead costs induced by the involvement of a UCD expert.

A. Context of the experiment

The method used consists in a retrospective and qualitative analysis of two software development projects: the first one without a usability expert in the team (to challenge hypotheses H1 and H3), the second one with one UCD expert in the team (to challenge hypotheses H2 and H3). The observations will help us to better define the issues related to *who should take the responsibility of the usability expert in Agile-UX*? Both observed projects are instantiations of Agile-UX.

They aim to develop a mobile application prototype, in order to demonstrate the interest of mobile touch-based

applications for the construction of site-related activities. The implemented prototype allows taking photos located on a construction site via a Global Positioning System (GPS. The user can highlight parts of a photo (e.g., add an arrow on a wall defect) and add textual or vocal notes about the entire photo or about the highlighted parts on the photo. The user can also register some construction sites by indicating their location on a map. Then, the photos are automatically attached to a construction site according to their location. The user can also find his photos in his calendar since the photos are automatically attached to events in his Google® calendar based on the shooting date. Finally, the user can share a set of photos with additional comments.

Two development projects were planned to experiment two different implementations of Agile-UX. Scrum was chosen as Agile method for both.

We chose small teams to facilitate this first observation and pay better attention to who does what, and what are the dynamics in the teams. The iteration durations were chosen by each team, according to the time deemed necessary by them to work at a convenient pace. The parallel tracks process from Sy was chosen as process in the second experiment because it was already used by a part of the team. We let the team choose the UCD methods they needed and the way to implement them (when and how). Dynamics throughout the projects have been observed, but usability of developed software was measured only at the end of the projects, in order to do not introduce a bias (e.g., like a competition between the teams).

1) Case #1 – Agile-UX without UCD expert: the first case studied does not involve a usability expert, so UCD is done by the team and particularly by the developer.

a) Composition of the team: In the first case, the team was composed of: a full-time developer, a Scrum master (part-time) and a business expert (part-time) who plays the role of product owner, and who is a researcher and an architect, with knowledge of architects' practices in France and Luxembourg.

All members of the team are aware to and have some knowledge in Human Computer Interaction (HCI) thanks to either an initial education that included courses on HCI or a business expertise.

b) Implementation of the UCD: The first case lasted six months with iterations' duration of one week. The team implemented Agile-UX on 22 iterations.

2) Case #2 - Agile-UX involving a usability expert: the second case studied involves a usability expert in the team, who is in charge of the deployment of the needed UCD methods.

a) Composition of the team: During the second case, the team was composed of a full-time usability expert (with an initial education on psychology and ergonomics), a full-time developer, a business expert (part-time) as product owner, and a Scrum master (part-time). The business expert and the Scrum master have either an initial education that included courses on HCI or a business expertise. The developer has neither particular awarness nor initial knowledge in HCI and, of course, he did not participate in case #1.

c) Implementation of the UCD: This development lasted six months with iterations of two weeks. Due to calendar constraints, the developer first started to work on technical requirements one month before the usability expert. For independent reasons, the usability expert quit the project before the end of the six months. The complete team only worked together for two and a half months. The process followed was the parallel track proposed by Sy [33].

B. Evaluation method

Different collecting methods and measure variables have been used to compare both projects, in order to challenge our hypotheses. They are presented below.

1) Quality of the product: The quality of the product developed by each team was measured with a focus on usability. The usability of each project has been measured by identifying the usability issues met by users, their number and their importance, but also by measuring the satisfaction of users.

Usability issues: Usability issues are problems encountered by users during the use of the software. For instance, they do not find the right button to perform an action, they are lost, some functionalities are missing, etc. Usability issues are raised thanks to users' tests. During users' test, users were asked to realize some scenarios, like find all photos taken during the last meeting. Errors made by users and their comments are observed, and are reformulated as usability issues. To define the importance of these usability issues, we propose the following formula see (1). Importance (i) represents the number of users who encountered the problem (n), multiplied by a seriousness indicator (s) stating whether a user was not able to pursue the interaction (from maximum level 4) or whether it was just a detail that did not impede the interaction (to minimum level 1).

$$i = n * s. \quad (1)$$

To evaluate the seriousness (or resolve's priority) of a usability problem, a decision tree inspired by the one defined by Cooper and Harper [11] was used. The decision tree (see Figure 4) enables to evaluate a seriousness level taking into account the importance of errors, their frequency and the users' success in fulfilling their tasks relative to the defined scenario.

Thus, to compare the quality of product of both projects, the number of problems met by users and the importance of problems raised were measured.

To complete this measure, we measure also the users' satisfaction.

a) Users' satisfaction: The users' satisfaction is defined as *"the users' success in fulfilling their tasks relative to the scenario"* in [18]. Several questionnaires exist to measure this satisfaction. We decided to use the

SUS questionnaire that was known by at least one person in each project team [8]. The questionnaire enables to calculate a percentage of satisfaction by user when using the software. The questionnaire was distributed to users at the end of their users' test.

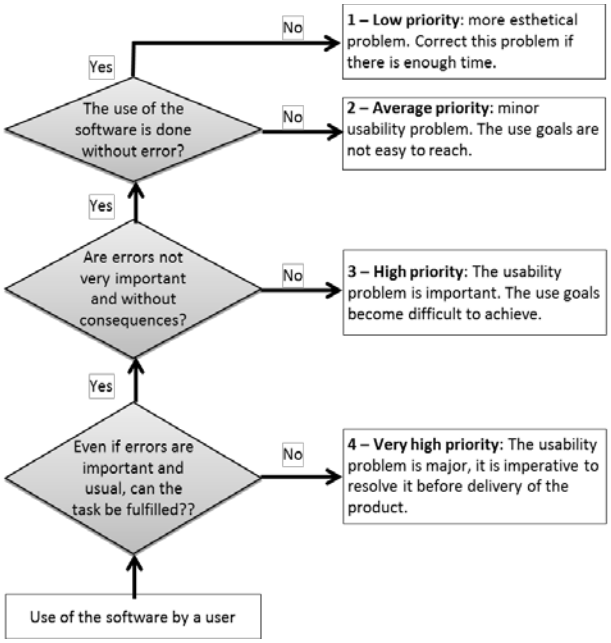


Figure 4. Decision tree to evaluate the seriousness (s) of a usability problem.

2) Team dynamics and satisfaction: Team dynamics and satisfaction of the teams were observe thanks to interviews of each team member throughout the projects. We participated to all meetings of each team and we observed occasionally some work sessions.

C. Results

Results of the experiment are discussed in terms of the quality of the final product, with a focus on the usability, the implemented UCD methods and the observed team dynamics. Usability is defined in ISO 9241-210 as *"[usability is the] extent to which a system, product or service can be used by specified users to achieve goals with effectiveness, efficiency and satisfaction in a specified context of use.* [18]" Some screen shots of the application developed in the first and second project are shown below (see Figure 5 and Figure 6).

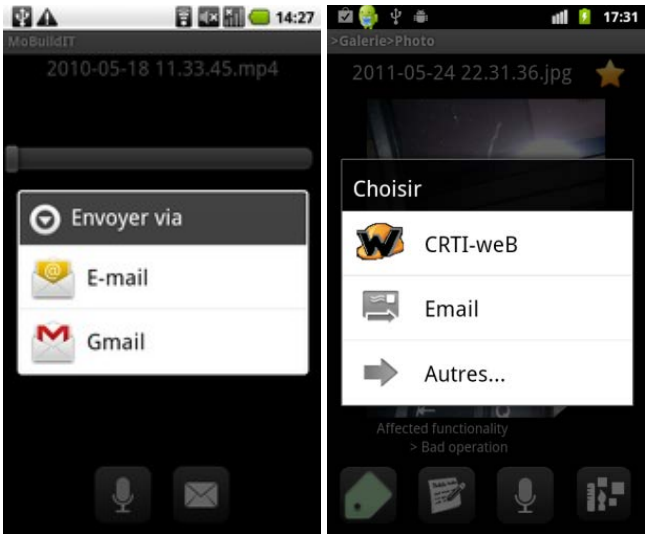


Figure 5. Photo sharing in first (left) and second project (right).

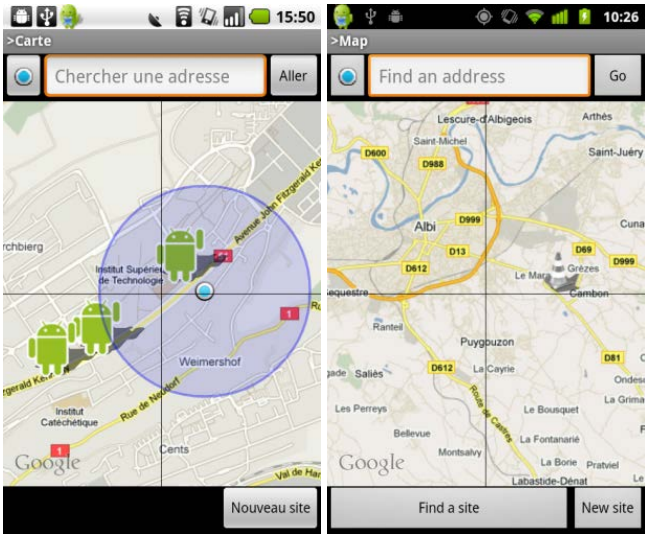


Figure 6. Site location in first (left) and second project (right).

1) Quality of the final product: challenging hypotheses H1 and H2, the quality of the product produced by each team with a focus on usability has been measured.

The users' tests raised fifteen problems encountered by users in the first case and only seven problems in the second one (see Table IV). Furthermore, problems are more important in the first case (11 problems with an importance between 1 and 8, and 4 problems with an importance between 10 and 20) than in the second one (7 problems with importance between 1 and 8 and no problem with importance higher than 8).

TABLE IV. USERS' TESTS RESULTS IN BOTH PROJECTS.

		Number of problems met	
		Use case 1	*Use case 2*
Importance of the problems (*i*)	1	5	2
	2	2	1
	3	3	1
	4	0	1
	6	1	1
	8	0	1
	10	1	0
	12	1	0
	15	1	0
	20	1	0
TOTAL		15	7

Users' satisfaction was measured thanks to the SUS questionnaire answered by users who made the users' tests [8]. It shows a lower average satisfaction in the first case (75.42%) than in the second one (81.25%). It can also be noted that extreme values are lower in the first case than in the second one (see Table V).

TABLE V. USERS' SATISFACTION RESULTS.

Percentage of users' satisfaction	Use case 1	Use case 2
Average	75.42 %	81.25%
Min	62.5 %	75%
Max	90 %	92.5%
Median	75	78,75
Deviation	8,74	6,73

2) Team dynamics and satisfaction: We used direct observations and interviews with team members in order to obtain a qualitative feedback on the team dynamics and satisfaction of team members on the interactions in the team.

Case #1: During this project, the developer played the role of UCD expert and developer of the application. As the developer had to play both roles, he had the feeling to progress slowly. Moreover, it is not easy to evaluate one's own work and to question it.

It should be noted that the team was in constant contact with the product owner thanks to his presence at every specification meeting, every demonstration meeting, and during some stand-up meetings. The product owner was also available to answer any team member's questions when necessary.

Case #2: The whole team had the feeling to progress quickly and to implement more functionalities, but also to obtain a better quality of the application.

Moreover, we observed the natural establishment of a *pair designing* [7][26]: when the developer was implementing wireframes, he sometimes asked the usability expert to join him and to explain and validate the developed interfaces during the implementation; when the usability expert designed wireframes, she sometimes asked the developer to join her and to validate the feasibility of wireframes during their design. Even if the developer had no skill in HCI at the beginning, he learnt the good practices throughout the project and quickly integrated them.

Furthermore, the team was in constant contact with the product owner through the Agile's dedicated meetings and also on demand.

3) Methods used: We also observed the UCD methods used in both projects. More UCD methods were deployed in the second case than in the first one, and both teams used different methods.

a) Case #1: The developer implemented only four usability methods: brainstorming, wireframing, users' tests, and satisfaction questionnaire. The following methods have been implemented by the team:

- Brainstorming sessions including business experts and technical experts to build the product backlog.
- Wire framing with Microsoft Power Point®.
- Two users' tests:
  - Real context of use, one user, one week
  - 6 architects, 6 scenarios, observation tests in laboratory. The results of these users' tests are presented in the previous section in Table IV.

b) Case #2: The following methods have been implemented by the UCD expert:

- Brainstorming sessions including business experts and technical experts to build the first version of the product backlog.
- Personas that help to define needs more precisely and improve the product backlog.
- Wire framing using paper & pen or sometimes Balsamiq®.
- Expert review based on ergonomics criteria (e.g., 10 usability heuristics for user interface design of Nielsen [24] or ergonomic criteria for the evaluation of human-computer interfaces of Bastien and Scapin [5]) after each release.
- Users' tests with four users: two users who know the application, and two novices. The results of these user tests are presented in the previous section, in Table IV.
- Focus groups to evaluate wire framing.

D. Discussion

Both projects' contexts and their results are synthesized in the Table VI. In the following, the hypotheses are discussed with regard to this experiment' results. Even if the results can be hardly generalized because of the very small sample size, our hypotheses tend to be confirmed.

1) Hypothesis 1: Results of users' tests and satisfaction questionnaire in case number 1 show that the users had encountered some problems. These problems are not very numerous (17) and their importance is relatively low (11 of the 17 problems noted have an importance inferior or equal to 8, with a maximum importance of 20). Satisfaction is quite good with an average satisfaction of 75.42%. With

regard to the first experiment results, Agile-UX works without a usability expert when some awarness and knowledge in HCI are available in the team. This justifies our first hypothesis H1.

TABLE VI. COMPARATIVE TABLE OF BOTH PROJECTS.

		Use case 1	Use case 2
Team	Developer	1 full-time	1 full-time
	Scrum master	1 part-time	1 part-time
	Product owner	1 part-time, business expert	1 part-time, business expert
	Usability expert	No	1 full-time
	Awarness to UCD	All team members	All team members, except the developer
Organisation of work	Duration	6 months	Expected 6 months – but 2,5 months
	Iteration duration	1 week	2 weeks
	Number of iterations	22	5
	Process	Scrum	Scrum + Sy's parallel tracks
UCD methods	Wire framing	Power Point®	Paper and pen + Balsamiq®
	Users' tests in direct observation	6 users, 6 scenarios	At every iteration end with 2 users who know the application and 2 novices
	Users' tests in real situation	1 user during 1 week	No
	Satisfaction questionnaire	Yes: SUS	Yes: SUS
	Personas	No	Yes
	Expert review	No	Yes, with Bastien and Scapin criteria
	Focus groups	No	To evaluate the wireframes
Other methods used	Brainstorming	To build the product backlog	To build the product backlog
Team dynamic and satisfaction	Feelings of the team	Slow progression	• Quick progression • Go further in the functionalities proposed • Improve quality of the application
	Observed team dynamic	No real dynamic Discouragement	• Pair-designing • Developer increased his HCI skills
Results		A lot of usability issues but working software.	• Lower number of usability issues identified by users and they are less critical. • Better users' satisfaction • And working software.

2) Hypothesis 2: We can detect that HCI skills of all team members would help avoid some usability mistakes in case 1, but, as the test results have shown, usability issues were identified by users. The test results show a lower number of usability issues in the second case (7 usability issues in the second experiment instead of 15 usability issues in the first experiment), thanks to the integration of the usability expert. It can also be noted that the usability issues are less critical in the second case than in the first case (see Table IV). Furthermore, the satisfaction of users, which is *correct* (75.42%) in the first case, is better in the second one (81.25%, see Table V). This justifies our second hypothesis H2: Agile-UX provides better quality in terms of usability with the involvement of a usability expert. This could also be explained by the different number and types of UCD methods used in both cases. In fact, in case 2, more methods were used because the usability expert was better trained and knew a wider variety of methods, but also because she had more time to dedicate to the deployment of these methods. Thus, an involvement of a usability expert in Agile-UX enables to use more methods, maybe more adapted, certainly best mastered.

3) Hypothesis 3: Without involving a usability expert, we observe a discouragement, particularly of the developer. On the contrary, involving a usability expert helps to maintain a constant pace in the team (principle 8 in [2]). No difference has been observed on the constant customer collaboration (value 3 in [1]). Some best practices emerged in the second case like *pair-designing*, and the whole team improved their practices and knowledge concerning HCI. This could justify our third hypothesis H3: the dynamic in the project team is better with a usability expert involved in Agile-UX. However, the fact that in the first case, the team was composed of only one person (the developer) may be of influence. Indeed, in the second case the team was composed of two people (the usability expert and the developer). Then, the dynamic observed may be due to the edge effect of the number of people in the team or simply to the personality of the people involved.

V. CONCLUSION AND FUTURE WORK

The literature review shows that the reconciliation of Agile and UCD is not a new trend, and it also shows that a number of research questions arise that have not been resolved today.

In this paper, we have further investigated the question of the necessity to involve a UCD expert in an Agile-UX team to support the UCD process. The state of the art shows that different types of involvement of UCD experts have been tried through different use cases, but the necessity of their involvement is neither justified nor discussed, and past experiments do not state the quality improvement implied by the involvement of UCD experts. To discuss this point, we proposed an experiment.

This experiment addressed two kinds of Agile-UX implementations. The first project did not involve a UCD expert in the team, but a team member, who was a generalizing specialist on development and on UCD was in charge of UCD. The second project involved a UCD expert in the team. With the help of these projects' observation, our two first initial hypotheses have been checked (H1-without usability expert, if the project team has awareness and some knowledge in HCI, Agile-UX gives a correct quality level about the product's usability; and H2-with usability expert involved in the project team, usability of the produced product is better than in H1). The third one cannot be checked at this step, even though observation shows that this hypothesis seems true (H3-the dynamic of the project team is better when a usability expert is involved). Further studies have to be conducted to have more quantitative results and to check the third hypothesis. Notably, the experiment protocol can be improved by evaluating also the usability of the developed software thanks to objective criteria like Nielsen usability heuristics [24] or Bastien and Scapin ergonomics criteria [5]. To that end, software are reviewed by usability experts thanks to one criteria guide. Usability issues can be grouped by kind of non-respecting criteria. An importance can be also calculated for each issue as we did for usability issues met by users during users' tests. During the experiment presented in this paper, all latitude was let to teams to select the UCD methods to use. The team 1 did not use this kind of expert review driven by objective criteria. Whereas team 2 chose to conduct at least one expert evaluation. It would be interesting to do a final expert evaluation to better compare objectively both results.

Future experiments will enable to measure the quality, in terms of usability, of software developed in Agile-UX. Completed projects that cover the different following variations will be selected:

- Number of UCD experts involved in the project, from zero to as many as there are developers involved in the team.
- Involvement modality of the person in charge of UCD: as a team member, as an external provider of services or as the product owner.

For each couple of parameters, at least ten projects with larger teams than the experiment presented above will be selected. The usability of selected completed projects will be measured through users' tests, satisfaction questionnaire and final heuristic evaluations driven by objective criteria (e.g., Nielsen criteria [24] or Bastien and Scapin ergonomics criteria [5]). These results will enable us to conclude what is the involvement modality that produces more usable software. In addition, the UCD methods deployed, their implementation mode (when and how), and the organization of work between developers and people in charge of UCD will be observed. Such observations would enable to identify good practices.

Another possible implementation of Agile-UX, which can be found in literature, is to place the usability expert as the product owner. In fact, the product owner is responsible for the contact with users, the definition of needs and the validation of the work done. A priori, some of the high level responsibilities of both, the usability expert and the product owner, overlap. A future task will be to check the legitimacy of the following hypothesis: the UCD expert could play the role of product owner.

Acknowledgments

The author thanks members of the CRAI laboratory of Nancy and the project's extended team members: Sylvain Kubicki, Annie Guerriero, Fabrice Absil, Marion Zéler, Luc Caffard and Charles Gilbertz. The author also would like to thank Jocelyn Aubert, Muriel Foulonneau. This paper is dedicated to Marion Zéler.

References

[1] L. Schwartz, "Agile-user experience design: with or without a usability expert in the team?," In Proc. ICSEA 2013, pp. 359-363.

[2] A. Alliance, "Agile manifesto," 2001, http://www.agilemanifesto.org, last access 2014.11.12.

[3] S. Ambler and M. Lines, "Disciplined agile delivery: A practitioner's guide to agile software delivery in the enterprise," 2012, IBM Press.

[4] J. Armitage, "Are Agile methods good for design?," Interactions, vol. 11, no 1, 2004, pp. 14-23.

[5] C. Bastien and D. Scapin. "Ergonomic criteria for the evaluation of human-computer interfaces," RT-0156, 1993, p.79.

[6] K. Beck, "Extreme Programming Explained: Embrace Change," Addison-Wesley Professional, 2003, p. 54.

[7] S. Blomkvist, "Towards a model for bridging Agile development and user-centered design," In Human-Centered Software Engineering—Integrating Usability in the Software Development Lifecycle, 2005, pp. 219-244, Springer Netherlands.

[8] J. Brooke, "SUS-A quick and dirty usability scale. Usability evaluation in industry," 1996, vol. 189, p.194.

[9] S. Chamberlain, H. Sharp, and N. Maiden, "Towards a framework for integrating Agile development and user-centred design," In Extreme Programming and Agile Processes in Software Engineering, Springer Berlin Heidelberg, Oulu, Finland, June 2006, pp. 143-153.

[10] L. Constantine, "Process agility and software usability: toward lightweight usage-centered design," Software, IEEE, vol 19, no. 2, 2002, pp. 42-50.

[11] G. E. Cooper and R. P. Harper, "The use of pilot rating in the evaluation of aircraft handling qualities," No. Agard-567, Advisory group for aerospace research and development, Neuilly-Sur-Seine, FRANCE, 1969.

[12] T. Dybå and T. Dingsøyr, "Empirical studies of Agile software development: A systematic review," Information and software technology, 2008, vol. 50, no. 9, pp. 833-859.

[13] J. Ferreira, J. Noble, and R. Biddle, "Agile development iterations and UI design," In Agile Conference (AGILE), Washington, DC, August 2007, pp. 50-58, IEEE.

[14] J. Ferreira, H. Sharp, and H. Robinson, "User experience design and Agile development: managing cooperation through articulation work," Software: Practice and Experience, vol. 41, no. 9, Wiley Online Library, 2011, pp. 963-974.

[15] D. Fox, J. Sillito, and F. Maurer, "Agile Methods and User-Centered Design: How These Two Methodologies Are Being Successfully Integrated In Industry," In Proc. Agile 2008, IEEE, 2008, pp. 63-72.

[16] J. Haikara, "Usability in Agile Software Development: Extending the Interaction Design Process with Personas Approach," Agile Processes in Software Engineering and Extreme Programming, Springer, 2007, pp. 153-156.

[17] ISO 16982:2002, "Ergonomics of human-system interaction -- Usability methods supporting human-centred design," International Standards for Business, Government and Society.

[18] ISO 9241-210:2010, "Ergonomics of human-system interaction -- Part 210: Human-centred design for interactive systems," International Standards for Business, Government and Society.

[19] D. Kane, "Finding a place for discount usability engineering in Agile development: throwing down the gauntlet," in Proceedings of the Agile development conference, 2003, pp. 40-46.

[20] M. Lindvall et al., "Empirical findings in Agile methods, Extreme Programming and Agile Methods," XP/Agile Universe 2002, Springer, 2002, pp. 81-92.

[21] P. McInerney and F. Maurer, "UCD in Agile projects: dream team or odd couple?," Interactions, vol. 12, no. 6, 2005, pp. 19-23.

[22] L. Miller, "Case Study of Customer Input For a Successful Product," in Proceedings of the Agile Development Conference, July 2005, pp.225-234.

[23] E. Nelson, "Extreme programming vs. interaction design," FTP Online, 2002.

[24] J. Nielsen, "Heuristic evaluation," In Nielsen, J. and Mack, R.L. (Eds.), Usability inspection methods, 1994, vol. 17, no. 1, pp. 25-62.

[25] C. Nodder and J. Nielsen, "Agile usability: best practices for user experience on Agile development projects," Nielsen Norman Group, 2010.

[26] A. Nummiaho, "User-Centered Design and Extreme Programming," In Software Engineering Seminar, Fall 2006, pp. 1-5.

[27] J. Patton, "Hitting the target: adding interaction design to agile software develop," In OOPSLA 2002 Practitioners Reports, 2002, ACM.

[28] D. Rawsthorne and D. Shimp, "Scrum In A Nutshell," SCRUM alliance, http://www.scrumalliance.org/articles/151-scrum-in-a-nutshell, last access 2014.11.12.

[29] L. Schwartz, "Agile-User Experience Design: an Agile and User-Centered Process?," In ICSEA 2013, October 2013, pp. 346-351.

[30] T. S. D. Silva, M. S. Silveira, and F. Maurer, (2013, August). "Ten Lessons Learned from Integrating Interaction Design and Agile Development," In Proc. AGILE'13, Nashville, TN, August 2013, pp. 42-49, IEEE.

[31] LM. Singh, "U-SCRUM: An Agile methodology for promoting usability," In Proc. AGILE'08, Tonronto, ON, August 2011, pp. 555-560, IEEE.

[32] J. Sutherland and K. Schwaber, Scrum, "The scrum papers: Nuts, bolts, and origins of an Agile process," 2007.

[33] D. Sy, "Adapting usability investigations for Agile user-centered design," Journal of usability Studies, vol. 2, no. 3, 2007, pp. 112-132.

[34] A. Y. Wale-Kolade, P. A. Nielsen, and T. Päivärinta, (2014). "Integrating Usability Practices into Agile Development: A Case Study," In V. Strahonja, N. Vrček, D. Plantak Vukovac, C. Barry, M. Lang, H. Linger, & C. Schneider (Eds.), Information Systems Development: Transforming Organisations and Society through Information Systems (ISD2014 Proceedings), Croatia, September 2014.

Reuse-Based Test Traceability: Automatic Linking of Test Cases and Requirements

Thomas Noack, Thomas Karbe
Technische Universität Berlin,
Daimler Center for Automotive IT Innovations (DCAITI)
Berlin, Germany
Email: {thomas.noack,thomas.karbe}@dcaiti.com

Steffen Helke
Brandenburg University of Technology
Cottbus - Senftenberg (BTU)
Cottbus, Germany
Email: steffen.helke@b-tu.de

***Abstract*—Safety standards demand full requirement traceability, which includes a complete tracing between requirements and test cases to stipulate how a requirement has to be verified. However, implementing such a concept rigorously is time-consuming and costly. Furthermore, in the automotive industry this cost is repeatedly incurred for each vehicle series, because in contrast to other development artefacts, reuse strategies for trace links have not yet been sufficiently researched. This paper presents the novel approach of Reuse-based Test Traceability, which allows for a more cost-effective implementation of trace links in certain cases. First, we identify and formalize a scenario, the so called RT-Problem, for reusing trace links between test cases and reused requirements, which has been observed in industry practice. Next, based on this formalization we propose a 3-layered method, which automatically creates links between test cases and reused requirements. For reasons of practicality, we focus on the first layer, which represents a transitive test-link reuse. Finally, we present the results of two field studies demonstrating that our approach is feasible in practice. As the main contribution of this work we show that the automated reuse of test cases on the basis of reused requirements is both possible and useful.**

Keywords–Reuse; Requirements; Test cases; Traceability.

I. MOTIVATION

New safety standards like ISO 26262 mean demand for traceability is higher than ever. Consequently, automotive companies must work hard to establish traceability for every phase in the V-Model. For instance, if a software error occurs, the specific part of the source code that has caused it should be identified. This is achieved by trace linking development artefacts. In hierarchical development processes, links between requirements and test cases are some of the first to arise. These links are an integral part of a relationship network. For example, an error is discovered by a test case. This test case is trace linked with a system requirement, which in turn is connected to the source code. Each kind of comprehensibility necessitates links between the requirements involved.

However, rigorously implementing such a concept is time-consuming and costly. Furthermore, in the automotive industry this cost must be paid for each vehicle series project, repeatedly, because in contrast to other development artefacts, reuse strategies that generate trace links automatically have not been sufficiently researched.

Therefore, we propose a novel method for reuse-based traceability, which extends [1] and allows for a more cost-effective implementation of trace links in certain cases.

Among other things, the ISO standard defines a demand for two categories of trace links. The first is called *Test Traceability* – ISO 26262 Pt. 8 [2, p.25] – and relates to a link convention for test specifications. Each specification in a test case must include a reference to the version of the associated work product. The second category relates to *Reuse-based Traceability* – ISO 26262 Pt. 6 [3, p.20] – which demands that every safety-related software component must be classified according to information on reuse and modification. Thus, the standard defines four classes: newly developed, reused with modification, reused without modification, and a commercial off-the-shelf product.

Our approach contributes to both claims. First, we provide a cost-effective technique for automatically generating trace links between test cases and requirements, which addresses the test traceability of the ISO standard. Secondly, we provide the generation of trace links between requirements or test cases from a previous project and the corresponding counterparts in a new project, to indicate that previous artefacts have been reused. Furthermore, our framework allows for these links to be qualified, by using types reflecting whether an artefact was modified or not.

Structure. The next section introduces a motivating example, illustrating a scenario where trace links can be reused. The section also gives important definitions of development artefacts. The section closes with the presentation of the RT-Problem (Reuse-based Test Traceability Problem), which forms the basis of this paper. Section III gives pointers to related work. We briefly survey existing traceability models and methods. We also identify limitations of these approaches to justify the need for this work. Section IV presents our 3-layered method, where we focus on the first layer, the so-called RT-linking technique. Section V describes theoretical concepts behind our RT-linking strategy. Subsequently, Section VI presents the results of two field studies demonstrating that our approach is feasible in practice. We also compare our technique with the previous manual procedure. To give the reader an impression of the complete method, Sections VII and VIII sketch the second and the third layer. The paper closes with conclusions and future work in Section IX.

II. System requirements and test cases

A. Introductory example

Figure 1 shows an example based on real specification documents. The upper left box contains artefacts from a previous vehicle series project, e.g., vehicle function *1000: Interrupt of front wiping during engine start*. Requirements *1001* and *1002* refine vehicle function *1000*. The lower box contains test cases, e.g., test case *5376: Washing during engine start*. Requirement *1002* and test case *5376* are connected via a trace link.

Requirements reuse. Reuse happens in all phases of the V-Model; therefore, reuse also applies to requirements. Requirements are reused from previous vehicle series in order to specify a new vehicle series. Technically, this reuse is achieved by copying and adapting the old requirements to the new vehicle series project. The upper right box in Figure 1 shows the reuse requirements. For example, the reuse requirement *1002-new* has been changed most: the description of the *washing interruption* has been moved.

Test trace reuse. This adaptation of requirements exemplifies the main feature of Reuse-based Test Traceability: Is test case *5376*, which already verifies source requirement *1002*, also suitable for verifying target requirement *1002-new*? More succinctly: Can test case *5376* be linked with target requirement *1002-new*?

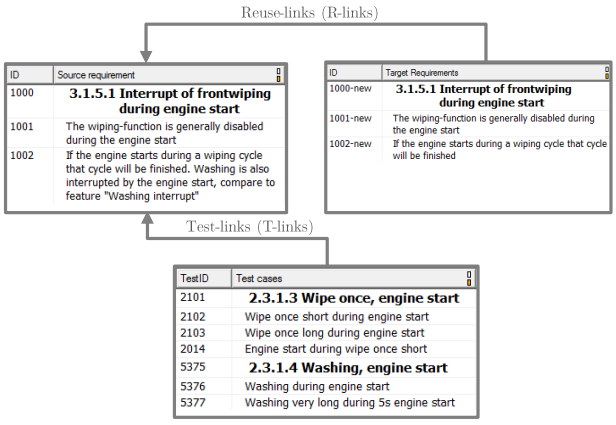


Figure 1: Example requirements and test cases in DOORS

B. System Requirements Specification (SRS)

A vehicle is described by many SRS. Every system, like the *Wiper Control* or the *Outside Light Control* is specified in one SRS. The main engineering artefacts in SRS are vehicle functions and the system requirements that refine them. Individual vehicle functions can be very extensive. For instance, the function *wipe windscreen* is characterized by several activation possibilities, wipe stages, passenger and pedestrian splash protection, etc., and therefore is refined by more than 300 requirements.

Count per vehicle series. If one function alone can have 300 requirements, how many requirements does a whole vehicle series have? The following estimate gives a vague idea: Modern vehicles have up to 100 electronic control units (ECU). Usually, multiple automotive systems run on each ECU. For simplicity, we assume that only one software system runs on each ECU. Further, we assume that midsize systems have at least 1000 requirements or more. Using these assumptions, a modern vehicle can accumulate hundreds of thousands or even millions of requirements.

Requirements classification. In addition, requirements have classifying properties such as Automotive Safety Integrity Levels (ASIL), testability, ownership, supplier status or dependencies to other SRS. Therefore, requirement categories exist, e.g., safety critical, testable or highly dependent requirements.

C. System Test Specification (STS)

Each SRS has at least one associated STS, which contains test cases to verify the correct implementation of the requirements. The structure of these test cases corresponds to the common schema: Pre-condition, post-condition, pass-condition, test steps etc. [4, p.263]. Each requirement which is classified as testable is trace linked to at least one test case. This facilitates comprehensibility to determine the requirements-based test case coverage.

Count per vehicle series. Again, the question arises: How many test cases does an entire vehicle series have? Because the test case count corresponds to the requirements count, a modern vehicle has at least as many test cases as requirements. Usually, more tests than requirements exist.

Test case classification. Like requirements, test cases have classifying properties such as test goals, test levels or test platforms. While test goals result from quality models as proposed in ISO 9126 [5], test levels describe the right branch of the V-Model. Automotive-specific test platforms include *Vehicle Network* or *HiL* (Hardware-in-the-Loop).

D. Test Concept (TC)

The ISO 26262 dictates the existence of a TC. It defines which test objects must be tested at which test level on which test platform in order to fulfil which quality goals (What? When? Where? Why?). The TC determines the relationship between the left and right branches of the V-Model. We focus on the system level of the V-Model because Reuse-base Test Traceability uses the *requirement* test object type. In our case, the TC defines which test cases must be trace linked with which requirements to sufficiently verify a vehicle series.

Usage of requirements and test case classification. The TC does not contain specific system requirements or test cases. It relies on the classifying properties of the artefacts involved. For instance, a requirement's ASIL ranking influences testing expenses because it is strongly related to the *test goal* and *test level* properties of the test case. The higher the ASIL ranking, the more test cases must be trace linked with requirements. The TC allows us to assess the trace link coverage between SRS and STS.

E. RT-*problem*

Reuse-based Test Traceability relies on a specific situation observed in industrial practice: The RT-problem. Figure 2 depicts how the RT-problem reflects the *Reuse* relationship between two SRS and the *Test* relationship between an STS and these two SRS. In the example, a requirement of the function *front wiping* (fw) from the SRS_{Src} has been reused by fw′ in the SRS_{Tgt}.

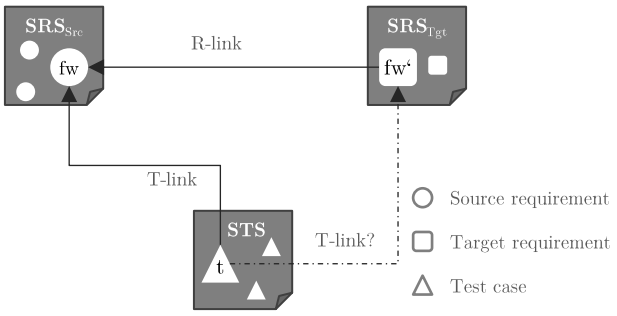


Figure 2: RT-problem within specification documents

R: Reuses. Among others, one reuse method is to copy an entire SRS into the project folder of the new vehicle series project. Thus two SRS result: the SRS_{Src} of the previous vehicle series and the adapted SRS_{Tgt} of the current vehicle series. Both SRS are in a reuse relationship: The SRS_{Tgt} reuses the SRS_{Src}.

T: Tests. Interestingly, the STS is not copied between vehicle series. Instead, the test cases are reused by redirecting trace links from the SRS_{Src} to the reusing SRS_{Tgt}. Overall, the RT-problem reflects the situation before the STS enters a test relationship with the SRS_{Tgt}.

R- and T-links. Technically, artefacts are connected by trace links. An R-link $fw' \rightarrow_R fw$ points from a target requirement fw′ to a source requirement fw. We also say (fw', fw) is a reuse pair. A T-link $t \rightarrow_T fw$ points from a test case t to a (source) requirement fw.

Solving the RT-problem. The RT-problem is unsolved if the dashed T-link in Figure 2 does not exist. In this paper, we propose transitive RT-linking, a new technique to set the T-link $t \rightarrow_T fw'$ to the target requirement fw′.

Business case. Earlier we estimated a whole vehicle series might have hundreds of thousands of requirements and test cases. Therefore, the RT-problem must be solved hundreds of thousands or even millions of times for each vehicle series project. Usually, test cases are T-linked during their creation. This T-linking requires little effort in comparison to the creation of the test case. STS are test case collections which are maintained over multiple vehicle series generations. In each new vehicle series project those STS must be linked manually. We observed that both time and motivation play an important role in why fewer and fewer T-Links exist from project to project. The primary goal of Reuse-based Test Traceability is that test cases only need to be T-linked with requirements once, at the time they are first written down.

III. Related work

A trace link connects trace artefacts and defines the type of relation between them [6, p.104]. In general, traceability is the possibility to establish and use trace links [7, p.9]. Thus, traceability enables comprehensibility. A traceability (information) model defines all possible artefacts and their types, as well as all possible links and their types [7, p.13]. Our work contributes to the special field of requirements traceability. Requirements traces are trace links between requirements and other software development artefacts [8, p.91]. Requirements traces always have a direction: forwards, backwards, inter or extra.

Link direction. A forward-trace connects a requirement with artefacts which have been created later in the development process. Examples of forward traces are links to architectural artefacts, source code or test cases. A backward-trace documents the origin of a requirement. Examples are links from laws or standards. An inter-trace links a requirement with another requirement. These links can reflect dependencies, refinement or even reuse. An extra-trace links a requirement with a non-requirement. Examples are architectural artefacts, source code or test cases.

Link direction in the RT-problem. An RT-problem consists of two links: an R-link and a T-link. The R-link connects a source and a target requirement. Therefore, it is an inter-trace link. Simultaneously, it is a backward-trace link because it reflects the origin of the target requirement. The T-link connects a test case with a source requirement. Thus, it is an extra- and forward-trace link. We observe that even though the RT-problem is very simple it contains inter, extra, forward and backward trace links.

A. *Traceability models*

Traceability models define the involved and linkable artefacts and the possible link types [9, p.106]. The RT-problem is a traceability model. The concept of traceability models appeared early in the development of software engineering: the first models appeared in the 1980s. The following paragraphs introduce relevant traceability models from the past 25 years.

General models without explicit T-links. The SODOS model [10] represents linkable artefacts via a relational database scheme. The trace links are freely configurable. Thus, SODOS is capable of connecting everything with everything. In the early 1990s, hypertext became very popular. The idea was to specify requirements and other software engineering artefacts by means of hypertext and link them using hyperlinks. Examples of hypertext traceability models include HYDRA [9], IBIS [11], REMAP [12], RETH [13], and the TOORS [14] model. Hyperlinks are mainly generic, so everything can be connected with everything else. Around the turn of the century, traceability models shifted their focus from hypertext models to UML-based traceability models [15] [16]. In accordance with the SOTA (State of the Art), older traceability models are more general while newer models are more specific. Newer work focuses on links between requirements [17] or links between requirements and design artefacts [18]. Although it is possible to define T-links in general traceability models, the models discussed here do not explicitly support T-links.

Models with T-links. In recent years, software testing has become increasingly popular. Thus, T-links have become an explicit part of traceability models. Ibrahim et al. propose the Total Traceability Model [19]. They consider requirements (R), test cases (T), design (D) and code (C). The model is exhaustive because it supports pair-wise extra-traces between the artefacts R-T, R-D, R-C, T-D, T-C and D-C. Furthermore, it supports the inter-traces D-D and C-C. Asuncion et al. have developed an End-to-End traceability model [20]. They consider marketing requirements (M), use cases (U), functional requirements (F) and test cases (T), which can be extra-linked in a M-U, U-F and F-T pipeline. Kirova et al. propose a traceability model, which uses performance requirements (PR), high level requirements (HLR), architectural requirements (AR), system requirements (SR), high level design (HLD), low level design (LLD), test cases (T) and test plans (TP). Kirova's model allows links between PR-AR, PR-SR, PR-T, HLR-SR, AR-SR, AR-T, AR-TP, AR-HLD, AR-LLD, SR-T, SR-TP, SR-HLD and SR-LLD. Azri and Ibrahim propose a metamodel [21] to allow trace links between arbitrary artefacts, including code, roles and even output files from developer tools.

RT-problem. The related work commonly draws holistic traceability pictures. Thus, a standard goal is to define holistic traceability models and exhaustively list the engineering artefacts and possible trace link types. However, the RT-problem is a specialized traceability model which focuses on a narrow set of circumstances. By using T-links explicitly, it focuses on the relationship between requirements and test cases. Additionally, the RT-problem introduces a new factor, the representation of requirements reuse with the help of R-links.

B. Traceability methods and techniques

The Requirements Traceability Matrix (RTM) was one of the first techniques that could systematically handle traceability [22]. The RTM is simply a table of requirement rows and linkable artefact columns. Each cell in the table represents a possible link. Requirements engineering tools like DOORS [23] support the RTM. Newer work is based on the idea of automatically creating trace links between artefacts and providing impact analysis. Two surveys [24, p.2] [25, p.31] categorize the SOTA of traceability methods as follows: event-based, rule-based, feature model-based, value-based, scenario-based, goal-based and information retrieval-based.

Event-based Traceability (EBT). EBT [26] introduces an event service where any linkable artefacts are registered. The service takes over the traceability and artefacts are no longer linked directly. Thus, EBT supports maintainability, as events trigger when artefacts change.

Rule-based Traceability (RBT). RBT [27] applies grammatical and lexical rules to find artefacts in structured specification documents and use case diagrams. A pairwise rule matching algorithm looks for artefacts, which match a rule. RBT links those related artefacts.

Feature Model-based Traceability (FBT). FBT [28] uses the feature as a connecting element between requirements and architecture as well as requirements and design. FBT uses several consistency criteria, e.g., whether each feature has at least one requirement and test case.

Value-based Traceability (VBT). VBT [29] assumes that a complete linkage between all involved artefacts is not feasible. Thus, VBT supports prioritized requirements and differently precise traceability schemes. The goal of VBT is to distinguish between links that generate benefits and links that only produce costs.

Scenario-based Traceability (SBT). SBT [30] uses scenarios, such as state chart paths, which are linked with requirements and code fragments. The creation of new links is performed transitively via code analysis. SBT is capable of completing links between requirements and code.

Goal-based Traceability (GBT). GBT [31] uses a quality model to define nonfunctional quality goals. Those nonfunctional goals are then connected to functional requirements. The goal of GBT is to trace the change impact from functional requirements to nonfunctional goals.

Information Retrieval-based Traceability (IBT). In recent years, IBT has become increasingly popular. Several approaches exist to finding and linking related artefacts [32]. The general idea behind IBT is to use information retrieval algorithms and similarity measures.

C. Alignment with SOTA (State-of-the-Art)

EBT propagates requirements change to linked artefacts. Thus, EBT is a technical event-based method to avoid manual tracing of impact and manual button clicks. Although it would be interesting to automatically execute the RT-problem solving technique after requirements reuse, we do not need EBT to solve the RT-problem methodically. RBT uses grammatical and lexical analysis to find similar requirements. Although it would be interesting to find reuse pairs with RBT, we assume that all R-links exist. Because of the importance of variability in the automotive domain, other research focuses on FBT [33]. VBT tries to reduce the number of links in order to reduce maintenance costs. The RT-problem is a specialized traceability model to reflect a simple circumstance and analyze it holistically. Thus, we do not want to remove any links in this way. SBT introduces an additional connecting artefact to link requirements and code. Because we focus on requirements and test cases, we do not desire new artefacts. Thus, SBT is also not suitable to solve the RT-problem.

New technique: RT-linking. Firstly, we propose a new technique to transitively link test cases by considering requirements reuse: RT-linking. We present a 3-layered method which integrates our RT-linking with parts of the SOTA. RT-linking is the primary method layer for creating new trace links between test cases and reusing requirements completely. To make the linking more precise we need additional filtering techniques, which are executed during the complete RT-linking. These filtering techniques define the other two layers of the 3-layered method. While the first layer establishes all links between test cases and reusing target requirements, the second and third layers filter out or highlight suspicious linking situations. The second layer uses the idea behind GBT to assess the test coverage with respect to test goals and other testing criteria. The third layer adapts IBT to search for similar RT-problems.

IV. 3-LAYERED METHOD

Figure 3 depicts the method layers as first proposed in [1]. Each layer consists of the same three phases. The specific tasks in each phase differ depending on the characteristics of the layer. Each subsequent layer enhances its predecessor's phases via additional tasks. The general tasks performed in the three phases are as follows:

- *Extract* RT-problems from SRS_{Src}, SRS_{Tgt} and STS.
- *Set/Filter* T-links from STS to SRS_{Tgt}.
- *Assess* T-links and highlight the link status.

Figure 4 depicts the 3-layered method in more detail. We will use the depicted example to briefly introduce the layers.

A. First layer: Transitive RT-Linking

Extract RT-problems. Figure 4 depicts an RT-problem: The target front wiping fw_{Tgt} reuses the source front wiping fw_{Src}. Thus, (fw_{Tgt}, fw_{Src}) is a reuse pair. The test case fw_{Test} has a T-link to fw_{Src}. Because fw_{Test} is not yet T-linked to fw_{Tgt}, one RT-problem is extracted.

Set T-links. The T-links of all extracted RT-problems are set transitively: If a test case is T-linked with a source requirement and this source requirement is R-linked with a target requirement, then the test case is also T-linked with the target requirement.

Assess T-links. Two scenarios can occur for each RT-linked target requirement: (a) It is textually identical to the source requirement and hence a T-link needs no review or (b) it has been changed and, therefore, the T-link must be reviewed manually. The third phase of each layer highlights the SRS_{Tgt} according to the assessment results.

B. Second layer: Test Concept-driven filtering

Extract RT-problems. Figure 4 indicates that the test case fw_{Test} meets the test goal *correctness of interfaces*. This information is extracted from the STS and appended to the test case of the RT-problem. We say that the RT-problem is augmented with the classifying property *test goal*.

Filter T-links. The TC defines whether a test case is needed to sufficiently verify a vehicle series. In Figure 4, the TC defines that the test goal *correctness of interfaces* must be met. Because the TC demands for a interface test case such as fw_{Test}, the RT-linking connects fw_{Test} and fw_{Tgt}. Otherwise, fw_{Test} would have been filtered out.

Assess T-links. While the first layer can only make statements about the existence of T-links, the second layer also considers the TC. Therefore, for each target requirement the following more detailed scenarios arise: (a) missing/superfluous test goals, (b) missing/superfluous test levels and (b) missing/superfluous test platforms. The SRS_{Tgt} is highlighted according to the TC coverage. TC-driven filtering has been proposed in [34].

C. Third layer: Case-Based filtering

Extract RT-problems. We assume that the requirements in Figure 4 have a classifying property, *interfaces*. While the source requirement fw_{Src} has an interface to the *column switch*, the target requirement has an additional interface to the *rain sensor*. In other words, the interfaces of the reuse pair (fw_{Tgt}, fw_{Src}) have changed. Again, we use the classifying properties to augment the extracted RT-problem.

Filter T-links. A Case Base contains RT-cases. RT-cases are RT-problems which include an RT-decision and a review note. The RT-decision defines whether a T-link can be set to a target requirement. Figure 4 depicts a simple RT-case: if the interfaces change, interface tests must be reviewed. This RT-case is very similar to our current RT-problem. Thus, the RT-decision *Review needed* is used to solve it.

Assess T-links. While the second layer only uses classifying test case properties, the third layer relies on fully augmented RT-problems. That means RT-cases can be defined freely by taking any classifying property into account. Thus, the assessment scenarios are as numerous as the RT-case possibilities. Finally, the RT-cases' review notes are copied into the SRS_{Tgt}.

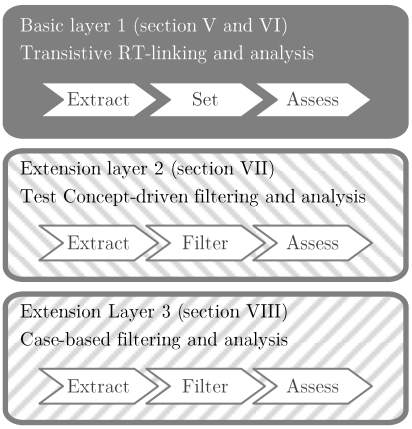


Figure 3: 3-layered method

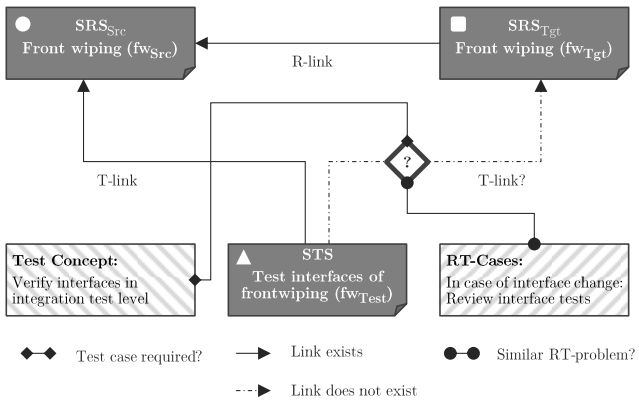


Figure 4: 3-layered method in more detail

V. RT-LINKING: CONCEPTS

A. Fundamental Example

Figure 1 showed an example with real requirements and test cases. However, real requirements and test cases are very extensive and also confidential. Therefore, the following sections will use the abstract example in Figure 5 to simplify the problem and concentrate on its relevant features.

Phase 1: Extract RT-Problems. Figure 5 shows the specification documents SRS_{Src}, STS and SRS_{Tgt}. Each example SRS contains one vehicle function, which is refined by requirements. While SRS_{Src} contains the source requirements src_i, SRS_{Tgt} contains the target requirements tgt_i. The *Reuse* column indicates for both SRS which source or target requirements the R-links point to. The *Test* column in SRS_{Src} indicates whether a source requirement is connected via a T-link with a test case. The *T-links: Before* column in STS displays which requirements the test cases were linked with before RT-linked them. Figure 5 contains three RT-problems with the following R- ($\rightarrow_R$) links and T-links ($\rightarrow_T$):

- $tgt_1 \rightarrow_R src_1$ and $test_1 \rightarrow_T src_1$,
- $tgt_2 \rightarrow_R src_2$ and $test_2 \rightarrow_T src_2$,
- $tgt_3 \rightarrow_R src_3$ and $test_{3/4} \rightarrow_T src_3$.

Phase 2: Execute RT-linking. The transitive RT-linking uses the following assumption:

IF	a target requirement reuses a source requirement
AND IF	a test case verifies a source requirement
THEN	the test case also verifies the target requirement.

After performing the RT-linking, the *T-Links: After* column contains the names of the source and target requirements which are linked with the test case by a T-Link. The *T?* column in SRS_{Tgt} indicates whether there is a T-link to a test case. The column is set to *Check* if the text for the target requirement has changed or if RT-inconsistencies (see **Phase 3**) have been uncovered. As Figure 5 shows, the second layer of the transitive RT-linking results in three solved RT-problems:

- $tgt_1 \rightarrow_R src_1$ and $test_1 \rightarrow_T src_1 \Rightarrow test_1 \rightarrow_T tgt_1$,
- $tgt_2 \rightarrow_R src_2$ and $test_2 \rightarrow_T src_2 \Rightarrow test_2 \rightarrow_T tgt_2$,
- $tgt_3 \rightarrow_R src_3$ and $test_{3/4} \rightarrow_T src_3 \Rightarrow test_{3/4} \rightarrow_T tgt_3$.

Phase 3: Assess T-links. The *similarity* column of SRS_{Tgt} represents the textual similarity of a reuse pair in percent. It is calculated with the help of well-known similarity measures, e.g., Dice, Jaro-Winkler or the Levenshtein distance [35]. As well as textual similarity, several other inconsistencies might occur, e.g., the SRS_{Src} describes a front and a rear wiper. A test case verifies both source requirements: If the reverse gear is engaged and if the column switch is pushed then the front and rear wipers will wipe. Now the target vehicle series does not provide a rear wiper. However, the SRS_{Tgt} reuses the front wiper requirement and the T-link from the test case. The inconsistency arises because the test case verifies a functionality (rear wiping) which does not exist in the target vehicle series.

Source System Requirement Specification (SRS_{Src})

Source requirements	Reuse	Test
1 Function Src		
src 1	tgt 1	Yes
src 2	tgt 2	Yes
src 3	tgt 3	Yes
src 4: will be discarded		Yes

System Test Specification (STS)

Test cases	T-links: before	T-links: after
1 Tests		
test 1	src 1	src 1, tgt 1
test 2	src 2	src 2, tgt 2
test 3/4	src 3 src 4	src 3, tgt 3 src 4

Target System Requirement Specification (SRS_{Tgt})

Target requirements	Reuse	Test	Similarity	Incons.	Review note
1 Function Tgt					
tgt 1: the same	src 1	Yes	100		
tgt 2: almost the same	src 2	Check	80		- Textual change
tgt 3: not the same	src 3	Check	60	II_Src	- Textual change - Test 3/4 links to Src 4 (discarded)
tgt 5: new		No			

Figure 5: Fundamental Before-After example

B. RT-*Linking*

The following basic sets describe the RT-problem:

$$SRS_{Src} \hat{=} \text{set of source system requirements}$$
$$SRS_{Tgt} \hat{=} \text{set of target system requirements}$$
$$STS \hat{=} \text{set of system test cases}$$

The upper left circle in Figure 6 represents SRS_{Src}, i.e., the set of all source system requirements. The upper right circle analogously represents SRS_{Tgt}, i.e., the set of all target system requirements. The lower circle shows the STS, i.e., the set of all system test cases. All three sets are disjoint, thus the overlapping areas in Figure 6 do not symbolize intersected sets, but linked elements between the disjoint sets.

Reuse pairs: R-links between requirements. An R-link $r_{tgt} \rightarrow_R r_{src}$ always points from target to source. A reuse target requirement from SRS_{Tgt} therefore points towards a reused source requirement from SRS_{Src}. Both target and source requirements can have multiple outgoing or incoming R-links. The following sets describe this information:

$$R := \{(r_{tgt}, r_{src}) \in SRS_{Tgt} \times SRS_{Src} \mid r_{tgt} \rightarrow_R r_{src}\}$$
$$R_{R,Src} := \{r_{src} \in SRS_{Src} \mid \exists r_{tgt} : (r_{tgt}, r_{src}) \in R\}$$
$$R_{R,Tgt} := \{r_{tgt} \in SRS_{Tgt} \mid \exists r_{src} : (r_{tgt}, r_{src}) \in R\}$$

The set R contains reuse pairs (r_{tgt}, r_{src}) which show that a R-link is pointing from r_{tgt} to r_{src}. Thus, $(r_{tgt}, r_{src}) \in R$ is a synonym for $r_{tgt} \rightarrow_R r_{src}$. The pair (r_{tgt}, r_{src}) is also a reuse pair. The sets $R_{R,Src}$ and $R_{R,Tgt}$ contain all requirements that are part of a reuse pair. Thus, $R_{R,Src}$ contains all reused source requirements from SRS_{Src}, and $R_{R,Tgt}$ contains all reusing target requirements from SRS_{Tgt}.

T-links from Test Cases to Requirements. A T-link $t \rightarrow_T r$ points from a test case to a requirement. An *active* test case points towards at least one requirement. A *verified* requirement has at least one incoming T-link from a test case. A test case can verify multiple source and/or target requirements, depending on whether it points into SRS_{Src}, into SRS_{Tgt} or both.

$$
\begin{aligned}
T &: SRS_{Src} \cup SRS_{Tgt} \rightarrow \mathcal{P}(STS) \\
T(r) &:= \{t \in STS \mid t \rightarrow_T r\} \\
R_{T,Src} &:= \{r_{src} \in SRS_{Src} \mid \exists t \in STS : t \rightarrow_T r_{src}\} \\
R_{T,Tgt} &:= \{r_{tgt} \in SRS_{Tgt} \mid \exists t \in STS : t \rightarrow_T r_{tgt}\}
\end{aligned}
$$

For a given requirement r (source or target), the function T derives all test cases that are linked with it. The set $R_{T,Src}$ contains all source requirements r_{src} from SRS_{Src} for which at least one T-link $t \rightarrow_T r_{src}$ points from a test case t from STS to r_{src}. Analogously, the set $R_{T,Tgt}$ contains all target requirements r_{tgt} that are linked with at least one test case t.

RT-linking. With these formal concepts we can reformulate the assumption for the transitive RT-linking:

IF	a target requirement r_{tgt} and
	a source requirement r_{src} are linked by an R-link
AND IF	a test case t is linked with r_{src} by a T-link
THEN	t can also be linked to r_{tgt} by a T-link.

This is represented by the formula

$$r_{tgt} \rightarrow_R r_{src} \wedge t \rightarrow_T r_{src} \Rightarrow t \rightarrow_T r_{tgt}$$

An RT-link, i.e., a solution for one of the three RT-problems from the abstract example in Figure 5 is shown here:

$$tgt_1 \rightarrow_R src_1 \text{ and } test_1 \rightarrow_T src_1 \Rightarrow test_1 \rightarrow_T tgt_1$$

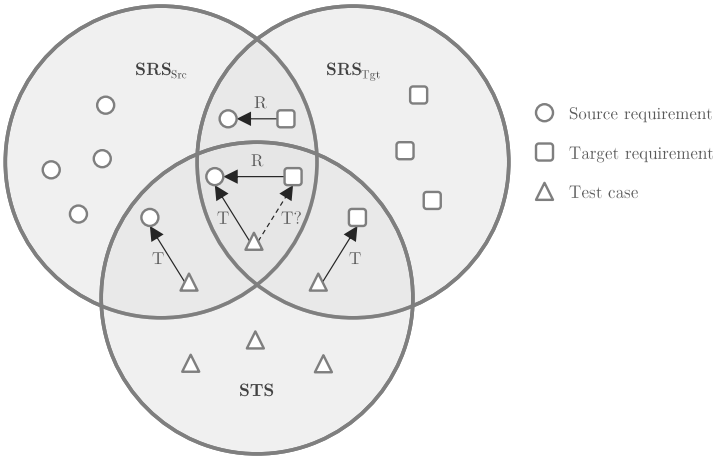


Figure 6: Each RT-problem is a RT-instance among others

C. RT-*diagram*

The RT-diagram is a means to represent the number of linking situations between requirements and test cases. It categorizes these situations into different types, e.g., reused but not tested requirements, reused and tested requirements. Figure 7 show the diagram schematically. The different segments of the diagram represent the different types of linking situations, which result from the existence or non-existence of links between the three basic sets SRS_{Src}, STS, and SRS_{Tgt}. Each segment is labelled with a different symbol (e.g., ○, □) that represents the type of the segment. In the industrial use and field studies in Section VI, the diagram segments show the number of linking situations.

RT-instances and RT-types. From now on, we call the different linking situations RT-*instances*. Every RT-instance is assigned to an RT-*type* (e.g., not tested but reused requirement). All RT-instances from the same segment of the diagram are also from the same RT-type. An RT-instance is denoted by a set, which contains either

- one artefact (meaning that this artefact is not linked)
  Corresponding types: ○, □, △,
- one link (meaning that the two linked artefacts are not linked to a third artefact)
  Corresponding types: ⊐, ⋒, ⋓,
- two links (meaning that one R-link and one T-link exists, but one T-link is missing to one of the partners of the reuse pair)
  Corresponding types: $\bullet_{inconsistent}$, or
- three links (fully linked, a solved RT-problem with a reuse pair and T-links to both partners of the pair)
  Corresponding types: $\bullet_{consistent}$.

On the next page we will examine the segments of the RT-diagram. Each segment will also be given a short name for future reference.

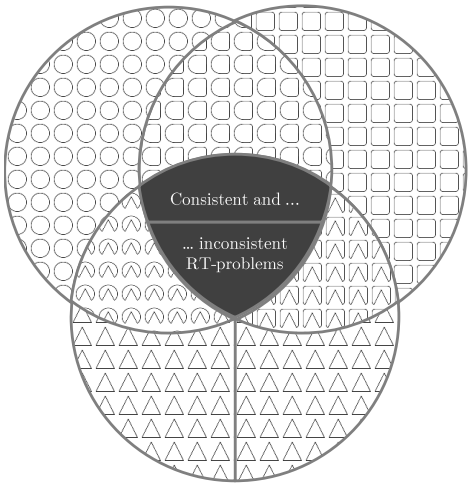


Figure 7: Types of RT-instances

Requirements without RT ($\bigcirc, \square$)**.** Figure 8a depicts those requirements which have no R-link and no T-link. The segment on the left-hand side represents all *discarded* SRS_{Src} requirements which have not been tested ($\bigcirc$). The segment on the right-hand side contains all *new* SRS_{Tgt} requirements which have no associated test cases ($\square$).

$$\bigcirc : \overline{R_{Src}}\,\overline{T_{Src}} := \{\{r_{src}\} \mid r_{src} \in SRS_{Src} \setminus (R_{T,Src} \cup R_{R,Src})\}$$
$$\square : \overline{R_{Tgt}}\,\overline{T_{Tgt}} := \{\{r_{tgt}\} \mid r_{tgt} \in SRS_{Tgt} \setminus (R_{T,Tgt} \cup R_{R,Tgt})\}$$

Reused requirements without T ($\square\!\!\square$)**.** Figure 8b depicts all reuse pairs (r_{tgt}, r_{src}), which have no T-link to either r_{tgt} or r_{src} . From the perspective of the SRS_{Tgt} these are all reusable and untested target requirements. Although we use the word *untested*, requirements with missing T-links do not remain untested in industrial practice. The mapping between requirements and test cases is then performed by engineers in an experience-based fashion. Of course, in this case testing takes place in SRS_{Tgt}; it is simply less traceable.

$$\square\!\!\square : R\,\overline{T_{Src*Tgt}} :=$$
$$\{\{r_{tgt} \rightarrow_R r_{src}\} \mid (r_{tgt}, r_{src}) \in R \wedge T(r_{tgt}) \cup T(r_{src}) = \emptyset\}$$

Tested requirements without R ($\mathbin{\reflectbox{$\ddots$}}, \ddots$)**.** Figure 8c shows all requirements that are not in a reuse relationship but which have associated test cases. The segment on the left-hand side depicts source requirements which are not reused, have no R-link but do have a T-link from a test case ($\mathbin{\reflectbox{$\ddots$}}$). The right-hand side shows all new target requirements which have no R-link but a new T-link ($\ddots$).

$$\mathbin{\reflectbox{$\ddots$}} : \overline{R_{Src}}\,T_{Src} := \{\{r_{src}\} \mid r_{src} \in R_{T,Src} \setminus R_{R,Src}\}$$
$$\ddots : \overline{R_{Tgt}}\,T_{Tgt} := \{\{r_{tgt}\} \mid r_{tgt} \in R_{T,Tgt} \setminus R_{R,Tgt}\}$$

RT-problems ($\blacktriangledown$)**.** Figure 8d shows the center of the RT-diagram. It represents the RT-problems, i.e., all RT-instances, which have a reuse pair (r_{tgt}, r_{src}) and a test case which is T-linked to at least one of the reuse partners. Therefore, the diagram center bundles three RT-types: RT-instances which have reuse pairs (r_{tgt}, r_{src}) and a T-link to r_{src} ($\blacktriangledown_{Src}$). RT-instances with reuse pairs which are only r_{tgt} T-linked ($\blacktriangledown_{Tgt}$). RT-instances which have reuse pairs and a T-link to both partners of the pair ($\blacktriangledown_{Both}$).

$$\blacktriangledown : R\,T_{Src+Tgt} :=$$
$$\left\{ \begin{array}{l} \{r_{tgt} \rightarrow_R r_{src}, t \rightarrow_T r_{src}\} \mid \\ (r_{tgt}, r_{src}) \in R \wedge t \in T(r_{src}) \wedge t \notin T(r_{tgt}) \end{array} \right\}$$
$$\cup \left\{ \begin{array}{l} \{r_{tgt} \rightarrow_R r_{src}, t \rightarrow_T r_{tgt}\} \mid \\ (r_{tgt}, r_{src}) \in R \wedge t \in T(r_{tgt}) \wedge t \notin T(r_{src}) \end{array} \right\}$$
$$\cup \left\{ \begin{array}{l} \{r_{tgt} \rightarrow_R r_{src}, t \rightarrow_T r_{src}, t \rightarrow_T r_{tgt}\} \mid \\ (r_{tgt}, r_{src}) \in R \wedge t \in T(r_{src}) \wedge t \in T(r_{tgt}) \end{array} \right\}$$

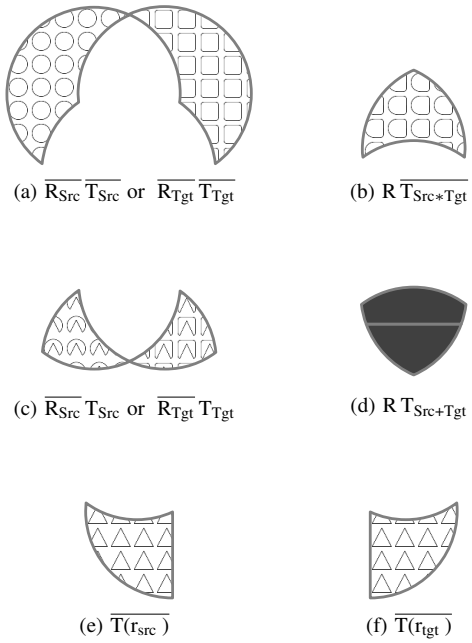
Test cases without T ($\Delta_{Src}, \Delta_{Tgt}$)**.** Figures 8e and 8f depict the test cases which have no T-links into SRS_{Src} or SRS_{Tgt}. In the context of the corresponding SRS those test cases are inactive.

$$\Delta_{Src} : \overline{T(r_{src})} := \{\{t\} \mid t \notin \bigcup_{r_{src} \in SRS_{Src}} T(r_{src})\}$$
$$\Delta_{Tgt} : \overline{T(r_{tgt})} := \{\{t\} \mid t \notin \bigcup_{r_{tgt} \in SRS_{Tgt}} T(r_{tgt})\}$$

Set of all RT-instances. An RT-instance represents a concrete link between one, two, or three artefacts. The set of all possible RT-instances RT_{Inst} is defined by:

$$RT_{Inst} := \bigcirc \cup \square \cup \square\!\!\square \cup \mathbin{\reflectbox{$\ddots$}} \cup \ddots \cup \blacktriangledown \cup \Delta_{Src} \cup \Delta_{Tgt}$$

The one-artefact instances $\{r_{src}\} \in \bigcirc$, $\{r_{tgt}\} \in \square$, $\{t\} \in \Delta_{Src}$ and $\{t\} \in \Delta_{Tgt}$ symbolize artefacts which are not linked to any other artefacts. The one-link instances $\{r_{tgt} \rightarrow_R r_{src}\} \in \square\!\!\square$, $\{t \rightarrow_T r_{src}\} \in \mathbin{\reflectbox{$\ddots$}}$, and $\{t \rightarrow_T r_{tgt}\} \in \ddots$ represent a link between exactly two artefacts, which both are not linked to any other artefact. The two-link instances $\{r_{tgt} \rightarrow_R r_{src}, t \rightarrow_T r_{src}\} \in \blacktriangledown_{Src}$, and $\{r_{tgt} \rightarrow_R r_{src}, t \rightarrow_T r_{tgt}\} \in \blacktriangledown_{Tgt}$ represent a source and a target requirement linked by an R-link and a test case which is linked to either the source requirement or the target requirement. However, the second test link is missing. The three-link instances $\{r_{tgt} \rightarrow_R r_{src}, t \rightarrow_T r_{src}, t \rightarrow_T r_{tgt}\} \in \blacktriangledown_{Src+Tgt}$ represent fully linked instances.



(a) $\overline{R_{Src}}\,\overline{T_{Src}}$ or $\overline{R_{Tgt}}\,\overline{T_{Tgt}}$ (b) $R\,\overline{T_{Src*Tgt}}$ (c) $\overline{R_{Src}}\,T_{Src}$ or $\overline{R_{Tgt}}\,T_{Tgt}$ (d) $R\,T_{Src+Tgt}$ (e) $\overline{T(r_{src})}$ (f) $\overline{T(r_{tgt})}$

Figure 8: Segments in the RT-diagram

D. RT-*inconsistencies*

We will now examine instances of the type $\bullet : R\, T_{Src+Tgt}$ in more detail. In these instances, test cases are T-linked with one or both partners of an RT-problem's reuse pair (r_{tgt}, r_{src}). But, in practice test cases are not only linked with one reuse pair. Usually, they have links to multiple reuse pairs or even requirements which have no reuse relationship to other requirements, which causes inconsistencies. Therefore, RT-inconsistencies are caused by missing T-links or unfavourable overlaps of reuse pairs with other requirements which are not part of a reuse pair, or by combining or splitting requirements. An RT-inconsistency is not necessarily an error, but an engineer should look into the RT-instance to check it. The following formulas describe the conditions for a reuse pair to be called *consistent*:

Consistency rule I. The first consistency rule says that a T-link must point to both partners in a reuse pair. If this rule is broken it indicates either overlooked source T-links or new target T-links. Two forms of rule I exist:

$$I_{Src}(r_{tgt}, r_{src}) := T(r_{src}) \subseteq T(r_{tgt})$$
$$I_{Tgt}(r_{tgt}, r_{src}) := T(r_{tgt}) \subseteq T(r_{src})$$

The first form I_{Src} says that the set of all test cases which are T-linked with the source requirement r_{src} of reuse pair (r_{tgt}, r_{src}) must also be T-linked with the target requirement r_{tgt}. The second inconsistency form I_{Tgt} is analogously defined: all test cases which are T-linked with r_{tgt} of a reuse pair (r_{tgt}, r_{src}) must also be T-linked with r_{src}.

Consistency rule II. While the first inconsistency rule assesses a reuse pair locally, the second inconsistency rule takes other requirements into account. It says that each test case that is T-linked to a given reuse pair is not allowed to test other requirements which are not part of another reuse pair. Therefore, this second rule highlights discarded source or newly added target functionality from the testing perspective. Again, two forms of inconsistency rule II exist:

$$\begin{aligned} II_{Src}(r_{tgt}, r_{src}) := \forall t \in T(r_{tgt}) \cup T(r_{src}) : \\ \forall r'_{src} \in SRS_{Src} : r'_{src} \neq r_{src} \Rightarrow \\ (t \rightarrow_T r'_{src} \Rightarrow \exists r'_{tgt} \in SRS_{Tgt} : \\ t \rightarrow_T r'_{tgt} \wedge (r'_{tgt}, r'_{src}) \in R) \end{aligned}$$

$$\begin{aligned} II_{Tgt}(r_{tgt}, r_{src}) := \forall t \in T(r_{tgt}) \cup T(r_{src}) : \\ \forall r'_{tgt} \in SRS_{Tgt} : r'_{tgt} \neq r_{tgt} \Rightarrow \\ (t \rightarrow_T r'_{tgt} \Rightarrow \exists r'_{src} \in SRS_{Src} : \\ t \rightarrow_T r'_{src} \wedge (r'_{tgt}, r'_{src}) \in R \end{aligned}$$

The first form II_{Src} says that a reuse pair (r_{tgt}, r_{src}) is consistent if all test cases of r_{src} are only T-linked with other source requirements r'_{src} which are part of a reuse pair (r'_{tgt}, r'_{src}), for some target requirement r'_{tgt}. In addition, all such test cases must be T-linked with r'_{tgt}. To improve the reade'rs understanding of inconsistency II_{Src} we will repeat the example of front and rear wipers. The front wiping requirement has been reused and the reuse pair $(front_{tgt}, front_{src})$ exists. Because the test case t is T-linked with $front_{src}$ it has been transitively RT-linked with $front_{tgt}$. Thus, t is T-linked with both partners of the front wiping reuse pair. Now, t is also T-linked with the source rear wiping requirement $rear_{src}$, which has not been reused because the new vehicle series does not provide any rear wiping. Since no reuse partner exists for $rear_{src}$, t causes $(front_{tgt}, front_{src})$ to be inconsistent II_{Src}: Test case t verifies functionality which does not exist in the target vehicle series. The second form II_{Tgt} is defined analogously from the perspective of the SRS_{Tgt}.

Consistency rule III. The third consistency rule indicates whether a target requirement has been amalgamated from multiple source requirements or a source requirement has been split into multiple target requirements. A T-link becomes problematic if a source requirement has been split. It is then unclear which target requirement needs to be T-linked. Again, two forms of consistency rule III exist:

$$\begin{aligned} III_{Src}(r_{tgt}, r_{src}) := \nexists r'_{tgt} \in SRS_{Tgt} : \\ r_{tgt} \neq r'_{tgt} \wedge (r'_{tgt}, r_{src}) \in R \end{aligned}$$

$$\begin{aligned} III_{Tgt}(r_{tgt}, r_{src}) := \nexists r'_{src} \in SRS_{Src} : \\ r_{src} \neq r'_{src} \wedge (r_{tgt}, r'_{src}) \in R \end{aligned}$$

The first form III_{Src} says that the source requirement r_{src} of the reuse pair (r_{tgt}, r_{src}) must not be a source partner of another reuse pair. III_{Tgt} says that the target requirement r_{tgt} of the reuse pair (r_{tgt}, r_{src}) must not be a target partner of another reuse pair.

Extension of the centre of the RT-diagram. The centre of the RT-diagram counts consistencies and inconsistencies. While consistently solved RT-problems are counted in the upper region of the centre, the inconsistent RT-instances are counted in the lower region. As depicted in Figure 9, we further differentiate between source and target inconsistencies in this lower centre region. Source inconsistencies X_{Src} indicate missing T-links which pointed to source requirements but not reusing target requirements or splits in source requirements. Target inconsistencies X_{Tgt} indicate progress because of newly added T-links. Source inconsistencies are bad inconsistencies and target inconsistencies are good inconsistencies.

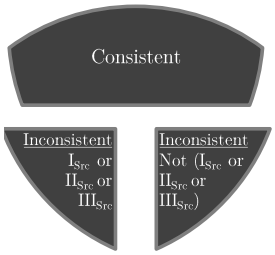


Figure 9: Inconsistencies in the diagram centre

Inconsistency I_{Src}. Figure 10a depicts the reuse pair (r_{tgt}, r_{src}) and the dashed T-link from test case t to r_{src}. The dashed T-link causes the reuse pair to be inconsistent I_{Src} because $T(r_{src}) \not\subseteq T(r_{src})$. This situation represents a typical RT-problem that needs do be solved.

Inconsistency I_{Tgt}. In Figure 10a, the reuse pair (r_{tgt}, r_{src}) would be inconsistent I_{Tgt} if the dashed T-link pointed from t to r_{tgt} instead of r_{src}, i.e., $t \rightarrow_T r_{tgt}$. In industrial practice, this situation usually occurs when a new test case has been added after reusing a requirement.

Inconsistency II_{Src}. Figure 10b depicts inconsistency II_{Src} in reuse pair (r_{tgt}, r_{src}). We clearly see that (r_{tgt}, r_{src}) is consistent I_{Src} and I_{Tgt} because $T(r_{src}) \subseteq T(r_{tgt})$ and $T(r_{tgt}) \subseteq T(r_{src})$. However, inconsistency II_{Src} for (r_{tgt}, r_{src}) is caused by the dashed T-link from t to r'_{src}. From the perspective of the reuse pair (r_{tgt}, r_{src}), test case t is T-linked with the requirement r'_{src}, which has no reuse pair partner and thus not has been reused.

Inconsistency II_{Tgt}. The reuse pair (r_{tgt}, r_{src}) in Figure 10b is inconsistent II_{Src}. It would be inconsistent II_{Tgt} if the dashed T-link pointed from t to a newly added r'_{tgt} instead of r'_{src}, which has not been reused.

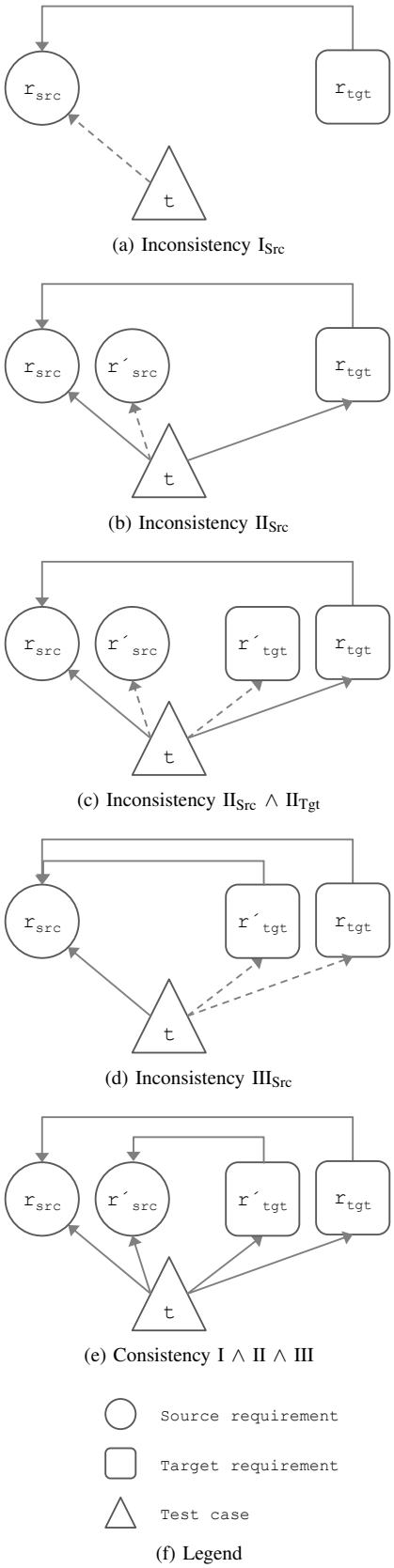
Inconsistency $II_{Src} \wedge II_{Tgt}$. Figure 10c depicts reuse pair (r_{tgt}, r_{src}), which is inconsistent II_{Src} and II_{Tgt}. Again, the reuse pair is consistent I_{Src} and I_{Tgt}. Test case t is T-linked with r'_{src} and r'_{tgt}, which are not a reuse pair. The left dashed T-link causes inconsistency II_{Src}, and the right dashed link causes inconsistency II_{Tgt}.

Inconsistency III_{Src}. Figure 10d depicts two reuse pairs, (r_{tgt}, r_{src}) and (r'_{tgt}, r_{src}), which are both consistent I_{Src} and I_{Tgt}. The source requirement r_{src} has been split into two target requirements r_{tgt} and r'_{tgt}. Test case t has been T-linked with both target requirements. Inconsistency III_{Src} occurs because source requirement r_{src} is partner of both reuse pairs.

Inconsistency III_{Tgt}. Figure 10d depicts III_{Tgt} if test case t was T-linked with all requirements of two reuse pairs, (r_{tgt}, r_{src}) and (r_{tgt}, r'_{src}).

Consistency. Figure 10e depicts two consistently T-linked reuse pairs, (r_{tgt}, r_{src}) and (r'_{tgt}, r'_{src}). Interestingly, inconsistency $II_{Src} \wedge II_{Tgt}$ from Figure 10c has been removed by adding the R-link $r'_{src} \rightarrow_R r'_{tgt}$. However, this is not a general solution, since both requirements are not necessarily partner of a reuse pair. However, this situation can still be used as an indicator to find forgotten R-links.

RT-Inconsistencies in the field. After laying the theoretical foundation for an analysis of RT-instances and RT-inconsistencies, we can now present the results of two field studies, thus showing the practical relevance of our RT-linking technique via real specification documents.



(a) Inconsistency I_{Src}

(b) Inconsistency II_{Src}

(c) Inconsistency $II_{Src} \wedge II_{Tgt}$

(d) Inconsistency III_{Src}

(e) Consistency $I \wedge II \wedge III$

(f) Legend

Figure 10: (In)Consistency examples

VI. RT-LINKING: FIELD STUDIES

From an industry perspective, the biggest advantage of the RT-linking is linking speed. However, the field studies will not focus on showing that automatic linking takes minutes instead of days (compared with manual linking). Instead, the primary goal is to show that RT-Linking is accurate.

A. Primary goal and preparation of the field study

RT-Linking is effective when it produces the same links as the current manual linking. Furthermore, the RT-linking is even more effective than the manual linking when it produces more T-links with fewer RT-inconsistencies.

Approach. In the past, RT-problems have been solved manually. We unsolve these historically solved RT-problems by removing all T-links that point to the target requirements. Next, we solve the RT-problems by RT-linking them again. The manually and automatically solved RT-problems will then be compared to assess the success of this field study.

Preparation. We were able to conduct these field studies in an industrial environment on real specification documents linked in an RT manner in IBM DOORS. These documents describe a historic linking situation between $SRS_{Src,Hist}$, $SRS_{Tgt,Hist}$ and STS_{Hist} of a past vehicle series' requirements and test reuse:

- The $SRS_{Src,Hist}$ contains source requirements.
- The $SRS_{Tgt,Hist}$ reuses $SRS_{Src,Hist}$.
- The STS_{Hist} contains test cases which point to $SRS_{Src,Hist}$ and $SRS_{Tgt,Hist}$.

The goal of the field studies is to show that automatic linking produces the same or even more T-links than the current manual linking. To achieve this, the historic documents were copied, including all links. After, all T-links between the copied STS_{RT} and the copied $SRS_{Tgt,RT}$ were removed. Thus, all documents only contained unsolved RT-problems:

- The $SRS_{Src,RT}$ is an unchanged copy of $SRS_{Src,Hist}$.
- The $SRS_{Tgt,RT}$ is a copy of $SRS_{Tgt,Hist}$. The copied SRS's have the same reuse pairs as the historical SRS's.
- The STS_{RT} is a copy of STS_{Hist} without any T-links into $SRS_{Tgt,RT}$. The T-links into $SRS_{Src,RT}$ remain.

Preparation: Special case. Before analysis of the overall linking situation can begin, we needed to reset all T-links which cause target inconsistencies. In the historic linking situation those T-links reflected new test cases which were T-linked with the $SRS_{Tgt,Hist}$ but not with the $SRS_{Src,Hist}$. We wanted the historic situation and the situation after the RT-linking to be comparable in terms of target inconsistencies. Therefore, we copied the T-links of all test cases which exclusively verified $SRS_{Tgt,Hist}$ into $SRS_{Tgt,RT}$. Those new test cases would have also been T-linked after RT-linking the artefacts from the copied documents.

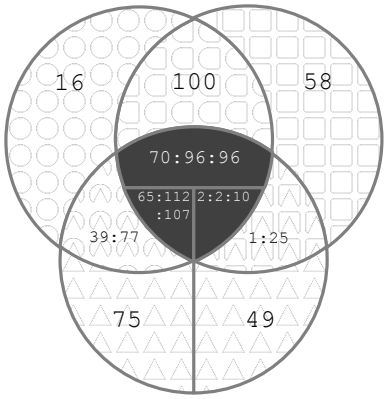
Execution. In order to compare the historic and automatic overall linking situations, all unsolved RT-problems were solved by automatically RT-linking them.

B. System A from vehicle series 1 to vehicle series 2

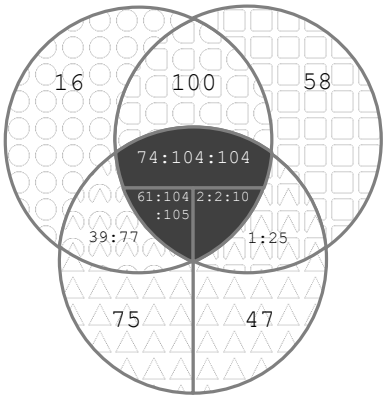
Figure 11 shows RT-diagrams to visualize the historic and RT-linking situation for a small-sized interior system.

Examination of the peripheral regions. RT-linking solves RT-problems. Because only the centre of the diagram represents RT-problems, automatic linking does not change most of the peripheral regions. Both diagrams show 16 source (○) and 58 target (□) requirements without R- and T-links. Both diagrams contain 100 reuse pairs without T-links (▢). Because the RT-linking does not change the T-links into the SRS_{Src}, both diagrams show 39 source requirements without R-links but with 77 T-links (⩓). In the field studies preparations we reset all exclusive T-links into the SRS_{Tgt}. Thus, both diagrams contain one target requirement without R-links but with 25 T-links (⩔). Both diagrams show that 75 test cases have no T-links into the SRS_{Src} ($\triangle_{Src}$). Only one peripheral region distinguishes the historic from the RT-linking situation: While 49 test cases have no T-links into $SRS_{Tgt,Hist}$, only 47 test case have no T-links into $SRS_{Tgt,RT}$ ($\triangle_{Tgt}$). This is a first clue that RT-linking is more efficient than the historic procedure.

Examination of the diagram centres. The diagram centres reveal that, historically, 70 RT-problems have been solved consistently while the RT-linking led to 74 consistently solved RT-problems. The following more detailed examination addresses this observation.



(a) Historic RT-diagram



(b) RT-diagramm after RT-linking

Figure 11: Overall linking situations

Detailed look at the inconsistencies. Table I represents the (in)consistently solved RT-problems within the diagram centres. The *Hist. situation* column represents the historic centre regions, while the *WvT linking* column stands for the centre regions of the RT-linking. The numbers R : T_{Src} : T_{Tgt} represent the count of reuse pairs (R), the count of T-links to the source requirements of the counted reuse pairs (T_{Src}) and the count of T-links to the target requirements (T_{Tgt}). Because R-links are a new concept they have been set automatically by matching document internal unique IDs of SRS_{Src} and SRS_{Tgt}. Thus, inconsistency III can not appear within the scope of the field study.

RT-diagrams and Table I. To increase the reliability of the field studies, two independent DXL scripts (DXL: Doors eXtension Language) were implemented. The first script calculates the numbers for the RT-diagram, while the second script calculates the numbers in Table I. The plausibility of the results is assured via the following rules: The numbers in the upper part of the diagram centres are the same as in row 0 of the table. The lower right region of the RT-diagram centres corresponds to the sum of the numbers in rows 2, 8 and 10. The lower left diagram centre's numbers correspond to the sum of all other table rows.

Table I: (In)consistencies in the diagram centers

(In)Consistency	Hist. situation	RT-linking
0: consistent	70:96:96	74:104:104
1: I_{Src}	1:5:4	-
2: I_{Tgt}	2:2:10	2:2:10
3: $I_{Src} \wedge I_{Tgt}$	-	-
4: II_{Src}	56:92:92	57:94:94
5: $I_{Src} \wedge II_{Src}$	4:5:0	-
6: $I_{Tgt} \wedge II_{Src}$	1:2:3	1:2:3
7: $I_{Src} \wedge I_{Tgt} \wedge II_{Src}$	-	-
8: II_{Tgt}	-	-
9: $I_{Src} \wedge II_{Tgt}$	-	-
10: $I_{Tgt} \wedge II_{Tgt}$	-	-
11: $I_{Src} \wedge I_{Tgt} \wedge II_{Tgt}$	-	-
12: $II_{Src} \wedge II_{Tgt}$	3:8:8	3:8:8
13: $I_{Src} \wedge II_{Src} \wedge II_{Tgt}$	-	-
14: $I_{Tgt} \wedge II_{Src} \wedge II_{Tgt}$	-	-
15: $I_{Src} \wedge I_{Tgt} \wedge II_{Src} \wedge II_{Tgt}$	-	-

RT-linking is effective. RT-linking is effective if at least the same test cases are automatically T-linked with $SRS_{Tgt,RT}$ as historically were T-linked with $SRS_{Tgt,Hist}$. First, the effectiveness is shown by the test cases not T-linked in the lower peripheral regions of both RT-diagrams. A simple for loop over all those test cases confirms that each test case which was not T-linked automatically was also not T-linked historically. Because fewer test cases have no T-links into $SRS_{Tgt,RT}$ after the RT-linking, the following statement is true: No test cases exist which have historical T-links into $SRS_{Tgt,Hist}$ but no automatically set T-links into $SRS_{Tgt,RT}$. Thus, the RT-linking is effective. A second for loop over all RT-instances in Table I confirms that each solved RT-problem in the *Hist. situation* column is also contained in the RT-*linking* column with the same or more T-links. Because at least the same T-links exist, RT-linking is effective. The existence of more T-links already indicates that RT-linking is even more effective than the current linking procedure.

RT-linking is even more effective. RT-linking is more effective than the current procedure if more test cases are T-linked and fewer RT-inconsistencies appear. A look at the diagram centres reveals that after the automatic RT-linking, 47 test cases do not point into the $SRS_{Tgt,RT}$. Thus, two fewer test cases are not T-linked in comparison to the historic linking situation. Given that the linking is effective, this leads to the conclusion that the RT-linking is even more effective than the current procedure. A further look at Table I confirms that more consistently solved RT-problems exist after the automatic linking. Row 0 of *Hist. situation* counts 70 reuse pairs. The source and target requirements of those 70 reuse pairs each count 96 consistent T-links. After the automatic RT-linking, 74 reuse pairs were counted. The source and target requirements are connected consistently with test cases for every 104 T-links. In conclusion, RT-linking results in more T-links and less RT-inconsistencies. Thus, RT-linking is more effective than the current manual procedure.

Origin of new consistencies. Thus far the field study has revealed that the proposed method solves more RT-problems consistently. The comparison of the centres of the two RT-diagrams showed that the upper centre region shows four additional consistently solved RT-problems, while the lower left centre region shows four fewer inconsistently solved RT-problems. Row 0 of Table I validated the plausibility of the observations. Now a question arises: From where did these new consistencies originate? We answer this question in the following sections by analysing the inconsistency transitions shown in Table II.

Transition rules (TR). Each reuse pair was analysed twice: once historically and once after the RT-linking. Therefore, we know, which inconsistency a reuse pair had in both cases. This enables us to compare the inconsistency transitions of each reuse pair. In the following, we will identify which inconsistency transitions exist and give examples for each transition that occurred. First, a summary of the four inconsistency rules: Consistency remains, inconsistency I_{Src} is eliminated, inconsistency II_{Src} remains or is eliminated, inconsistencies I_{Tgt} and II_{Tgt} remain. These rules can describe all observed inconsistency transitions.

TR_1: Consistency remains. This first transition rule says that each historic RT-pair which is consistent, remains consistent after RT-linking it automatically with test cases. This was the case for all 70 consistent RT-pairs.

TR_2: Inconsistency I_{Src} is eliminated. RT-linking eliminates inconsistency I_{Src} by definition, because it transitively sets T-links between source and target requirements. Tables I and II confirm the elimination of I_{Src}, because they never contain inconsistencies with an odd number after the RT-linking (because I_{Src} denotes 2^0).

TR_3: Inconsistency II_{Src} remains or is eliminated. II_{Src} can become consistent if an inconsistency II_{Src} occurs, because an inconsistency I_{Src} occurred at another point. Otherwise II_{Src} remains. In Table II, historic inconsistencies II_{Src} remained ($4 \Rightarrow 4$, $6 \Rightarrow 6$, $12 \Rightarrow 12$) or were eliminated ($4 \Rightarrow 0$, $5 \Rightarrow 0$) by the RT-linking.

TR_4: Inconsistencies I_{Tgt} and II_{Tgt} remain. To enable a comparison between $SRS_{Tgt,Hist}$ and $SRS_{Tgt,RT}$ in preparation of the field study, all exclusive T-links into $SRS_{Tgt,Hist}$ were also set exclusively to $SRS_{Tgt,RT}$. Exclusively means that the T-link points into SRS_{Tgt} but not into SRS_{Src}. Target inconsistencies I_{Tgt} and II_{Tgt} caused by this show that new test cases have been T-linked with the SRS_{Tgt}. Because the linking of new test cases does not impact RT-linking, all target inconsistencies remain. Table II confirms this ($2 \Rightarrow 2$, $6 \Rightarrow 6$, $12 \Rightarrow 12$).

Table II: Inconsistency transitions from *Hist.* to RT

Hist. $\Rightarrow$	RT-linking	#Transitions
$0 \Rightarrow 0$	(consistent $\Rightarrow$ consistent)	70
$1 \Rightarrow 0$	($I_{Src} \Rightarrow$ consistent)	1
$2 \Rightarrow 2$	($I_{Tgt} \Rightarrow I_{Tgt}$)	2
$4 \Rightarrow 0$	($II_{Src} \Rightarrow$ consistent)	1
$4 \Rightarrow 4$	($II_{Src} \Rightarrow II_{Src}$)	55
$5 \Rightarrow 0$	($I_{Src} \wedge II_{Src} \Rightarrow$ consistent)	2
$5 \Rightarrow 4$	($I_{Src} \wedge II_{Src} \Rightarrow II_{Src}$)	2
$6 \Rightarrow 6$	($I_{Tgt} \wedge II_{Src} \Rightarrow I_{Tgt} \wedge II_{Src}$)	1
$12 \Rightarrow 12$	($II_{Src} \wedge II_{Tgt} \Rightarrow II_{Src} \wedge II_{Tgt}$)	3

Transition examples. To improve the reader's understanding of inconsistency transitions, an example will be presented for each row in Table II. The following figures show real but anonymised transitions. Therefore, some inconsistency transitions appear isolated in a single example while other examples contain multiple transitions. The figure on the left hand side always represents the historically solved RT-problem(s), while the right hand figure always shows the same RT-problem(s), only automatically RT-linked.

Transition: Consistency remains ($0 \Rightarrow 0$). All source and target requirements in Figure 12 build a reuse pair: (A, a), (B, b). The test case t is T-linked with all partners of all reuse pairs. Thus, TR_1 applies and consistency remains.

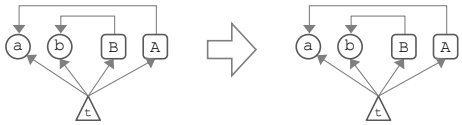


Figure 12: $0 \Rightarrow 0$

Transition: I_{Src} is eliminated ($1 \Rightarrow 0$). Figure 13 shows the reuse pair (A, a). On the left hand side the dashed T-link $t \rightarrow_T a$ causes the historic RT-problem to remain unsolved. This implies inconsistency I_{Src}. The definition of the RT-linking does not allow such situations. Rule TR_2 applies and the T-link $t \rightarrow_T A$ exists after performing the RT-linking.

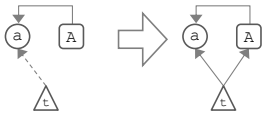


Figure 13: $1 \Rightarrow 0$

Transition: I_{Tgt} remains ($2 \Rightarrow 2$). Figure 14 depicts the reuse pair (A, a). Because $t \rightarrow_T A$ originates from the new test case t, $t \rightarrow_T a$ does not exist. This causes inconsistency I_{Tgt}, which remains according to TR_4.

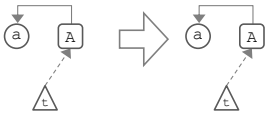


Figure 14: $2 \Rightarrow 2$

Transition: II_{Src} remains ($4 \Rightarrow 4$). Figure 15 depicts the reuse pair (B, b) and source requirement a, which has not been reused. Inconsistency II_{Src} is caused by $t \rightarrow_T a$ because t verifies functionality that does not exist on the target side. Rule TR_3 applies: II_{Src} cannot be eliminated, because it is not caused by another inconsistency I_{Src}.

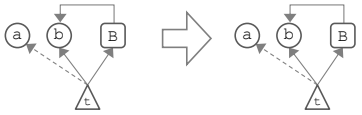


Figure 15: $4 \Rightarrow 4$

Transition: II_{Src} is eliminated ($4 \Rightarrow 0$). Figure 16 depicts several dependent RT-problems. The reuse pair (C, c) is consistent I_{Src} because t is T-linked to both partners. But the T-links t $\rightarrow_T$ a and t $\rightarrow_T$ b cause (C, c) to be inconsistent II_{Src}. Rule TR_2 applies. Thus, the inconsistencies I_{Src} of (A, a) and (B, b) are eliminated by setting t $\rightarrow_T$ A and t $\rightarrow_T$ B. Rule TR_3 applies: The pair (C, c) is now consistent because I_{Src} of (A, a) and (B, b) was the reason for its II_{Src}.

Transition: $I_{Src} \wedge II_{Src}$ becomes consistent ($5 \Rightarrow 0$). Figure 16 shows inconsistency II_{Src} is eliminated from the perspective of (C, c). Because the rules TR_2 and TR_3 also apply to the pairs (A, a) and (B, b) their inconsistencies $I_{Src} \wedge II_{Src}$ are eliminated.

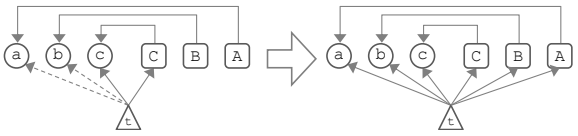


Figure 16: $4 \Rightarrow 0$ und $5 \Rightarrow 0$

Transition: $I_{Src} \wedge II_{Src}$ becomes II_{Src} ($5 \Rightarrow 4$). Figure 17 depicts the reuse pair (B, b) and source requirement a, which has not been reused. Rule TR_2 applies for (C, c): The missing T-link t $\rightarrow_T$ B is set via the RT-linking. Inconsistency I_{Src} for (B, b) is caused by t $\rightarrow_T$ a. Therefore, I_{Src} is not eliminated by RT-linking and TR_3 applies: (C, c) remains II_{Src}.

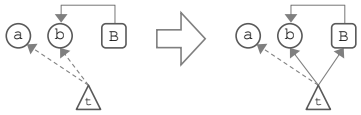


Figure 17: $5 \Rightarrow 4$

Transition: $I_{Tgt} \wedge II_{Src}$ remains ($6 \Rightarrow 6$). Figure 18 shows reuse pair (B, b) and the source requirement a, which was not reused. The T-link t $\rightarrow_T$ B causes (B, b) to be inconsistent I_{Tgt}. Rule TR_4 forces I_{Tgt} to remain. Additionally, TR_3 applies: (C, c) remains II_{Src} because of t $\rightarrow_T$ a.

Transition: $II_{Src} \wedge II_{Tgt}$ stays ($12 \Rightarrow 12$). Figure 18 also shows the reuse pair (C, c). Again, the T-link t $\rightarrow_T$ a causes (C, c) to be unresolvably inconsistent II_{Src} and TR_3 applies. Additionally, TR_4 applies and IITgt remains.

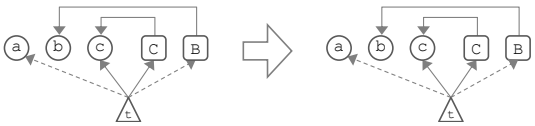


Figure 18: $6 \Rightarrow 6$ und $12 \Rightarrow 12$

C. System B from vehicle series 3 to vehicle series 1

The goal of the first field study was to show that RT-linking is at least as effective as the current manual procedure. To support a full and detailed analysis of the results, the field study was performed on a small-sized system A. The second field study serves another purpose: confirmation.

Confirmation. The second field study was performed on a bigger system B, which had five times more requirements and test cases than system A. The field study was conducted using the same preparations as the first field study. All inconsistency rules were confirmed. No new rules appeared, but more inconsistency transitions did. All new transitions can be described by the four basic inconsistency rules, TR_1, TR_2, TR_3 and TR_4.

D. RT-linking: Conclusion

In Section V, we proposed the basic layer of our 3-layered method to automatically link test cases with reusing requirements. RT-linking uses the assumption:

IF	a target requirement reuses a source requirement
AND IF	a test case verifies a source requirement
THEN	the test case also verifies the target requirement.

Extension of the SOTA. We gave a short introduction into research into requirements traceability in Section III. There we summarized different traceability methods (EBT, RBT, VBT, ...), which automatically create links between requirements and other artefacts. The proposed RT-linking extends the research field via a new method: Reuse-based Test Traceability (RTT).

Supporting industrial practice. RT-linking did not just arise from observing industry practice. We also evaluated the effectiveness of RT-linking under real circumstances with a DOORS extension plug-in. We conducted a field study to compare the overall linking situation of a set of specification documents after manual linking and automatic RT-linking. Thanks to its promising results in terms of accuracy, this plug-in has been transferred to industrial practice. In addition to its accuracy, the complete linking and link analysis of a SRS_{Tgt} now takes a few minutes instead of days or even weeks of manual link creation and maintenance. However, it is clear that some of the T-links have to be reviewed manually. Those T-links which can be established without review define the business case for RT-linking. Several pilot projects showed that these vary between 20% (systems of new vehicle series with many new/changed requirements) and 90% (systems of facelifts with little/no changes). Overall we can say that specification documents of future vehicle series projects will likely be RT-linked.

Outlook: Extension of the RT-linking. The main part of this paper focused on the primary contribution of this work: RT-linking. We will now provide an overview of extensions to RT-linking, as proposed in [1].

VII. Outlook: Test Concept-driven Filtering

RT-linking is the first method layer. It solves the RT-problem. However, real world testing necessitates further considerations. Systematic test planning satisfies both ISO 26262 and testing efficiency. Therefore, we introduce an augmentation to the RT-problem's test case.

Augmentation to the RT-problem. *Test goals* (What purpose?), *test levels* (When?) and *test platforms* (Where?) of the Test Concept [Section II-D] are used to determine the testing expenses. These test planning dimensions describe a three dimensional *testing expenses cube* for each vehicle function. The configuration of each cube defines the testing expenses for a vehicle function and thus also for their refining requirements. Figure 19 depicts the RT-problem, which is augmented with classifying test case properties. Each test case aims to meet specific test goals, is executed during specific test levels and is run on specific test platforms. These classifying test case properties are identical to the test planning dimensions of the *testing expenses cube*. Figure 19 depicts the concept behind the Test Concept-driven Filtering. Test case t is linked with target requirement r_{tgt} if the classifying properties of t fit the configuration of the cube. In this context, new RT-inconsistencies arise, for example: (1) If a test case meets more test goals than demanded by the cube for the vehicle function then too many test goals are met. Thus, the testing is not minimally efficient. (2) If the test cube demands more test goals then all T-linked test cases put together, then too few test goals will be met. As a result, the testing is not complete.

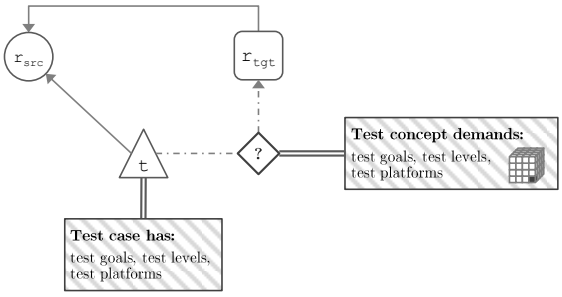


Figure 19: Test planning dimensions of the RT-problem

Extension of RT-linking. Test Concept-driven Filtering extends RT-linking as follows:

IF	a target requirement reuses a source requirement
AND IF	a test case verifies a source requirement
THEN	the test case also verifies the target requirement according to the Test Concept.

This second method layer has been proposed in [34]. Further, the filtering technique has been implemented in DOORS DXL. It is currently being evaluated in an industrial environment.

VIII. Outlook: Case-based Filtering

In the second method layer, we only augmented the test case of the RT-problem in order to connect it with the Test Concept. The third layer eliminates this unused potential by *fully* augmenting the RT-problem. In this way, we facilitate similarity between RT-problems.

Full augmentation of the RT-problem. Figure 19 depicts the test case augmentation with classifying test case properties. Figure 20 depicts the additional augmentation with classifying requirements properties. These requirements and test properties can be defined freely, e.g., ASIL ranking, interfaces, requirements maturity, OEM or supplier status, test case derivation method, ownership, etc. By virtue of the R-link between requirements, property changes can be detected from source to target in relation to the test properties. As the following example illustrates, this presents countless possibilities. We assume that the owner of a source requirement and a T-linked test case is *Thomas*. Further, we assume that the owner of the target requirement has changed to *Jonathan*. Case-based Filtering allows us to detect such situations. The detection mechanism adapts the retrieval techniques of Case-based reasoning as follows: An RT-case is structurally identical to an RT-problem, except for an additional RT-decision and review note. More precisely, Figure 20 does not depict an RT-problem but an RT-case. We use a similarity function to assess the similarity of the classifying properties between an RT-problem and an RT-case. Thus, we not only detect similar RT-problems for a given RT-case, but also adapt the RT-decision and review note to solve the RT-problem.

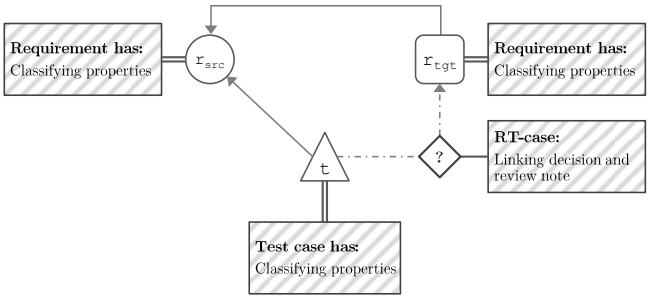


Figure 20: Classifying properties of the RT-problem

Extension of RT-linking. Case-based Filtering extends RT-linking as follows:

IF	a target requirement reuses a source requirement
AND IF	a test case verifies a source requirement
THEN	the test case also verifies the target requirement according to the RT-Case-basis.

A prototype of the third layer has been implemented in DOORS DXL. An initial presentation sparked the interest of requirements and test engineers.

IX. Conclusions and Future Work

In this paper, we proposed RT-linking, the main technique of our 3-layered method, to automatically solve the RT-problem. RT-problems occur after requirements reuse (R) but before test reuse (T). The main goal is to reuse test cases on the basis of requirements reuse.

Comprehensive requirements reuse. We proposed a new trace link type: The Reuse- or R-link, which connects reusing target requirements with reused source requirements. These R-links facilitate comprehensive reuse traceability as demanded by ISO 26262 [3, p.20]. By virtue of R-links we are able to systematically distinguish between new or discarded requirements and requirements which have been reused with or without modification.

Comprehensive test specification. Test cases are trace linked with requirements by Test- or T-links to facilitate test traceability according to ISO 26262 [2, p.25]. These T-links are influenced by the requirements reuse; reuse of requirements implies reuse of test cases. In this paper, we proposed transitive RT-linking to set T-links automatically from test cases to reusing target requirements on the basis of previously T-linked source requirements. Thanks to the promising results from the field studies, RT-linking has been transferred to industrial practice.

Comprehensive test planning and decisions. Although this paper focused on the first method layer, we gave a short overview of the second and third method layers. Both layers provide additional filter techniques to improve the RT-linking technique. The second layer arranges RT-linking according to test planning, while the third layer defines and uses similarity between RT-problems to reuse link decisions.

Rotate and tilt the RT-problem. Outside of industrial application, an interesting idea has arisen. We used requirements, test cases and specific link types to describe the RT-problem. However, it could be much more general than this if we used abstract artefacts and abstract link types instead. V-Models are usually characterized by many different trace linked artefacts. As well as requirements and test cases, systems, components, safety cases, architecture, feature models, functional models and code also exist, among others. Even the SOTA often relies on holistic traceability models [Section III-A]. Future work could focus on the interesting question of whether a general RT-problem is able to support a traceability model by successively rotating or tilting it through the V-Model, each followed by a general RT-linking step.

X. Acknowledgements

We thank our DCAITI colleagues Quang Minh Tran, Jonas Winkler and Martin Beckmann for discussions and proof reading. Furthermore, we thank our Daimler colleagues for positively impacting our concepts and implementations with respect to relevance, feasibility and usability.

References

[1] T. Noack, "Automatic Linking of Test Cases and Requirements," in Proceedings of the 5th International Conference in System Testing and Validation Lifecycle (VALID), Venice, Italy, 2013, pp. 45–48.

[2] Road Vehicles - Functional Safety, Part 8: Supporting processes (ISO 26262-8:2011), International Organization for Standardization, 2011.

[3] Road Vehicles - Functional Safety, Part 6: Product Development: Software Level (ISO 26262-6:2011), International Organization for Standardization, 2011.

[4] A. Spillner and T. Linz, Basiswissen Softwaretest, 4th ed. Heidelberg: dpunkt.verlag, 2005.

[5] Software Engineering - Product Quality, Part 1: Quality Model (ISO 9126-1:2001), European Committee for Standardization and International Organization for Standardization, 2005.

[6] R. Watkins and M. Neal, "Why and How of Requirements Tracing," IEEE Software, vol. 11, no. 4, 1994, pp. 104–106.

[7] J. Cleland-Huang, O. Gotel, and A. Zisman, Software and Systems Traceability, 1st ed. Springer, 2012.

[8] F. A. C. Pinheiro, "Requirements Traceability," in Perspectives on Software Requirements, 1st ed., J. C. S. do Prado Leite and J. H. Doorn, Eds. Kluwer Academic Publishers, 2004, ch. 5, pp. 91–113.

[9] K. Pohl and P. Haumer, "HYDRA: A Hypertext Model for Structuring Informal Requirements Representations," in Proceedings of the 2nd International Workshop on Requirements Engineering (REFSQ), K. Pohl and P. Peters, Eds. Jyväskylä, Finnland: Verlag der Augustinus-Buchhandlung, 1995, pp. 118–134.

[10] E. Horowitz and R. C. Williamson, "SODOS : A Software Documentation Support Environment - Its Definition," IEEE Transactions on Software Engineering, vol. 12, no. 8, 1986, pp. 849–859.

[11] J. Conklin and M. L. Begeman, "glBIS : A Hypertext Tool for Team Design Deliberation," in Proceedings of the ACM Conference on Hypertext. Chapel Hill, NC, USA: ACM Press, 1987, pp. 247–251.

[12] B. Ramesh and V. Dhar, "Supporting Systems Development by Capturing Deliberations During Requirements Engineering," IEEE Transactions on Software Engineering, vol. 18, no. 6, 1992, pp. 498–510.

[13] H. Kaindl, "The Missing Link in Requirements Engineering," ACM SIGSOFT Software Engineering Notes, vol. 18, no. 2, 1993, pp. 30–39.

[14] F. A. C. Pinheiro and J. A. Goguen, "An Object-Oriented Tool for Tracing Requirements," in Proceedings of the 2nd International Conference on Requirements Engineering. Colorado Springs, CO, USA: IEEE Computer Society Press, 1996, pp. 52–64.

[15] T. Tsumaki and Y. Morisawa, "A Framework of Requirements Tracing using UML," in Proceedings of 7th Asia-Pacific Software Engineering Conference (APSEC). Singapur, Singapur: IEEE Computer Society Press, 2000, pp. 206–213.

[16] P. Letelier, "A Framework for Requirements Traceability in UML-Based Projects," in Proceedings of 1st International Workshop on Traceability in Emerging Forms of Software Engineering, Edinburgh, Schottland, 2002, pp. 30–41.

[17] M. Narmanli, "A Business Rule Approach to Requirements Traceability," Masterarbeit, Middle East Technical University, 2010.

[18] B. Turban, M. Kucera, A. Tsakpinis, and C. Wolff, "Bridging the Requirements to Design Traceability Gap," Intelligent Technical Systems, Lecture Notes in Electrical Engineering, vol. 38, 2009, pp. 275–288.

[19] S. Ibrahim, M. Munro, and A. Deraman, "Implementing a Document-Based Requirements Traceability: A Case Study," in Proceedings of International Conference on Software Engineering, P. Kokol, Ed. Innsbruck, Österreich: ACTA Press, 2005, pp. 124–131.

[20] H. U. Asuncion, F. Francois, and R. N. Taylor, "An End-To-End Industrial Software Traceability Tool," in Proceedings of the 6th Joint Meeting of the European Software Engineering Conference and the ACM SIGSOFT Symposium on The Foundations of Software Engineering. Dubrovnik, Kroatien: ACM, 2007, pp. 115–124.

[21] A. Azmi and S. Ibrahim, "Implementing Test Management Traceability Model to Support Test Documents," International Journal of Digital Information and Wireless Communications (IJDIWC), vol. 1, no. 1, 2011, pp. 109–125.

[22] J. MacMillan and J. R. Vosburgh, "Software Quality Indicators," Scientific System Inc., Cambridge MA., Tech. Rep., 1986.

[23] DOORS (Dynamic Object Oriented Requirements System) , IBM, 2013.

[24] S. Rochimah, W. M. N. Wan Kadir, and A. H. Abdullah, "An Evaluation of Traceability Approaches to Support Software Evolution," in International Conference on Software Engineering Advances (ICSEA 2007). Cap Esterel, Frankreich: IEEE Computer Society, Aug. 2007, pp. 19–38.

[25] R. Torkar, T. Gorschek, R. Feldt, M. Svahnberg, U. A. Raja, and K. Kamran, "Requirements Traceability: A Systematic Review and Industry Case Study," International Journal of Software Engineering and Knowledge Engineering, vol. 22, no. 3, 2012, pp. 385–433.

[26] J. Cleland-Huang, C. K. Chang, and M. Christensen, "Event-Based Traceability for Managing Evolutionary Change," IEEE Transactions on Software Engineering, vol. 29, no. 9, 2003, pp. 796–810.

[27] G. Spanoudakis, A. Zisman, E. Pérez-Minana, and P. Krause, "Rule-Based Generation of Requirements Traceability Relations," Journal of Systems and Software, vol. 72, no. 2, Jul. 2004, pp. 105–127.

[28] M. Riebisch, "Supporting Evolutionary Development by Feature Models and Traceability Links," in 11th IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS). Brno, Tschechien: IEEE Computer Society Press, 2004, pp. 370–377.

[29] M. Heindl and S. Biffl, "A Case Study on Value-Based Requirements Tracing," in Proceedings of the 10th European Software Engineering Conference held jointly with 13th ACM SIGSOFT International Symposium on Foundations of Software Engineering. Lissabon, Portugal: ACM Press, 2005, pp. 60–69.

[30] A. Egyed and P. Grünbacher, "Supporting Software Understanding with Automated Requirements Traceability," International Journal of Software Engineering and Knowledge Engineering, vol. 15, no. 5, 2005, pp. 783–810.

[31] J. Cleland-Huang, R. Settimi, O. BenKhadra, E. Berezhanskaya, and S. Christina, "Goal-Centric Traceability for Managing Non-Functional Requirements," in Proceedings of the 27th International Conference on Software Engineering (ICSE). St. Louis, MO, USA: ACM Press, 2005, pp. 362–371.

[32] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia, "On the Equivalence of Information Retrieval Methods for Automated Traceability Link Recovery," in 18th IEEE International Conference on Program Comprehension (ICPC). Minho, Portugal: IEEE Computer Society Press, Jun. 2010, pp. 68–71.

[33] A. Cmyrev, R. Nörenberg, D. Hopp, and R. Reissing, "Consistency Checking of Feature Mapping between Requirements and Test Artefacts," Concurrent Engineering Approaches for Sustainable Product Development in a Multi-Disciplinary Environment, vol. 1, no. 1, 2013, pp. 121–132.

[34] T. Noack, "Automatische Verlinkung von Testfällen und Anforderungen: Testplangesteuerte Filterung," Softwaretechnik-Trends, vol. 33, no. 4, 2013, pp. 5–6.

[35] SimMetrics (Open Source library with numerous algorithms to calculate textual similarity between two texts), Source Forge, 2014.

An Analysis of the Implementation of Agile Software Development Practice in Irish Industry

Empirical research in a sample of Irish Industry

Trish O'Connell
Galway-Mayo Institute of Technology, Galway, Ireland.
trish.oconnell@gmit.ie
National University of Ireland, Galway, Ireland.
Department of Information Technology
trish.oconnell@nuigalway.ie

***Abstract*— On reading the vast amount of literature that has been written on Agile methodologies such as Scrum and XP one is invariably faced with a number of guidelines associated with successful implementation. The Agile Manifesto has stated various values and principles that need to be inculcated into any Agile software development undertaking. We might then expect that software development organisations that self-report as being Agile are abiding by the recommended precepts of their chosen Agile method, be this Scrum or Extreme Programming (XP). This paper presents empirical research that was undertaken in a sample of Irish software development organisations with a view to determining if the Agile precepts were being followed by organisations that self-described their software development process as either Scrum or XP.**

Keywords-Agile; Scrum; XP; Agile guidelines

I. INTRODUCTION

"In the absence of any *a priori* knowledge, it is generally believed that if companies claim to be Agile then they are, in fact, following the precepts and guidelines of their chosen Agile methodology" [1]. In terms of the chosen approach/methodology this could refer to Scrum [2], eXtreme Programming [3], Crystal Clear [4] or indeed "any of a plethora of Agile practices" [1].

Agile software development takes an iterative approach to developing software where feedback from stakeholders is combined with a team based approach to deliver software artifacts, which are of genuine value to the Customer. The salient principle is recognition of the unpredictability of the software development process and an acknowledgement that to overcome this, a flexible, quick-response methodology must be employed.

It would appear that software development organisations have embraced Agile development wholeheartedly. According to the 7th Annual State of Agile Development Survey [5], which was compiled in 2013, more than 84% of those surveyed claimed their organisations were practicing Agile development. Agile has been in vogue since 2001 with the launch of the Agile manifesto [6]. Even non-technical Customers have heard about Agile. It appears to have become a 'must have' in the same way as the ISO9000 quality management standard was a de rigueur requirement for enterprises to do business in the nineties. But for all of the organisations claiming to be Agile one must wonder as to whether it has simply become a buzzword that is used to allay the trepidations of Customers. Buglione [7] states, "Agile is become one of most known and used ICT buzzwords of last 10 years." Perhaps software development organisations are using the buzzword but neglecting to adhere to the 'spirit' of the Agile manifesto.

Ambler & Lions argue for a more disciplined approach to mainstream agile practices such as Scrum and Extreme Programming (XP), stating that "mainstream agile methods don't provide enough guidance for typical enterprises" [8].

With this in mind it was decided to conduct "some quantitative research into aspects of actual Agile implementation in a sample of Irish software industry with a view to gaining an understanding of the level of compliance to documented Agile precepts" [1].

This paper seeks to ascertain whether organisations that lay claim to being Agile, do, in fact, adhere to the acknowledged precepts and guidelines of the chosen Agile method or whether the implementation is more of an *ad hoc* approach, as the author believes to be the case.

Section II of this paper examines the background to Agile. Section III takes an in-depth look at two of the foremost Agile methodologies in use, Scrum (which is really an agile project management approach to software development) and Extreme Programming (XP). Section IV of this paper outlines the research that was conducted including a breakdown of the study methodology and the participants. Section V presents the results of the research and lead into Section VI, where the findings are presented. Section VII presents a discussion of the results and leads into Section VIII, which examines the limitations of the research and plans for future work. Finally, Section IX presents the conclusions.

II. LITERATURE REVIEW

Since the advent of software development there have been many different models proposed to improve the development and delivery of software to the Customer. One of the major issues faced by organisations is that most often Customer requirements are not clear at the outset or are misunderstood by the designers. Added to this the fact that

the requirements are rarely 'set in stone' means the designers are effectively trying to cope with constant flux or what Dooley [9] refers to as "software requirements churn." Developing software is not an easy task.

Set within this context there are the many diverse models for software development; some, like the Waterfall model and its derivatives are referred to as the Traditional models. They treat software development as a linear sequence of prescribed phases, which unfortunately does not always reflect the reality of developing a software product or system. Similarly, the Unified Process is also an iterative and incremental approach to software development. However, in the last decade a new paradigm has emerged called Agile software development.

The term Agile was coined in 2001 and whilst its true meaning relates to the ability of the software development organisation to adapt to requirements churn, i.e., to be agile with regard to change, the term Agile is also used as an umbrella term to describe a range of software development methodologies, which use an iterative approach to developing software. Cockburn [4] states "Agile processes can take on late-changing requirements exactly because of early and frequent delivery of running software, use of iterative and timeboxing techniques, continual attention to architecture, and willingness to update the design."

Whilst most know that Agile is an iterative approach to software development, which takes account of the changing requirements and lack of predictability that are inherent in a software project, the actual precepts of Agile might not be quite so well-known.

The traditional approach to software development was largely plan based insofar as a list of requirements were presented by the Customer at the outset and, once this was agreed to by the Developer, the Customer would normally sign off on a Requirements document. From then on, the development followed a sequential path in which discrete phases fed into each other, e.g., the requirements phase was followed by the architectural definition and so on until the product or system was delivered to the Customer.

However, requirements change and often what was delivered was not what was ultimately desired. For this reason it was acknowledged that there had to be a better way to develop software.

In 1970, Dr. Winston Royce presented a paper [10], in which he outlined the inherent risks in adopting a phased or sequential approach to software development. He argued that it was safer (in terms of minimizing risk) to use an iterative approach and also he suggested that it was "important to involve the Customer in a formal way so that he has committed himself at earlier points before final delivery" [10]. Paradoxically, the sequential approach to which Dr. Royce was referring has become what today is called the Waterfall method. It would appear that its proponents have missed the fact that Dr. Royce was using the model to describe what might be called, at best, a flawed model of software development.

In 2001, however, when seventeen key software developers (including Jim Highsmith, Martin Fowler, Mike Beedle, Ken Schwaber, Jeff Sutherland and Kent Beck to name but a few) met to discuss 'lightweight' development methods they agreed on what became referred to as the Agile Manifesto [6]. Thus, Agile promotes an adaptive approach to planning and an evolutionary, iterative approach to software development. Changes in requirements are both expected and managed and the Customer is seen as key to the successful outcome.

Unsurprisingly, each of the signatories of the Agile Manifesto had their own perception of what was agreed and consequently within the space of a few years there were "many different approaches to implementing Agile and each has its own 'vanilla' version" [1]. Sutherland [11] refers to "different approaches for implementing the core values from the Agile manifesto."

Section III of this paper now examines the fundamentals of two of the most used Agile methods: Scrum and Extreme Programming (XP). Before proceeding, however, it must be acknowledged that Scrum and XP can be thought of as complementary Agile methods given that Scrum is a product development methodology whereas XP is an engineering methodology.

III. AGILE METHODS

As previously stated, are many different approaches to implementing Agile and each has its own 'vanilla' version. Sutherland [11] explains, "Each Agile methodology has a slightly different approach for implementing the core values from the Agile Manifesto, just as many computer languages manifest the core features of object-oriented programming in different ways." The methodologies chosen for the study were Scrum and Extreme Programming (XP), since preliminary research into this domain identified these as the most prominent of the Agile methodologies currently in use for software development. Salo & Abrahamsson [12] refer to Scrum and XP as "perhaps best known agile methods." Scrum will be treated first.

A. SCRUM

The first use of the method that would become known as Scrum was by Ken Schwaber in the 1990s at his company Advanced Development Methods. At approximately the same time Jeff Sutherland, who was working at Easel Corporation, is credited as being the first to refer to the approach as Scrum. In 1995, the two developers jointly presented their Scrum methodology at OOPSLA 1995 [2]. The developers' contention was that software development's "linear nature has been its largest problem" [2]. Schwaber argued that the Waterfall process "does not define how to respond to unexpected output from any of the intermediate processes" [2]. The Spiral model was similarly criticized for "each of the phases consisting of linear, explicitly defined processes" [2]. The solution to the non-linearity of the software process was to acknowledge the software development process as "complicated and complex" [2]. Accordingly, a method which would allow teams of developers to "operate adaptively within a complex environment using imprecise processes" [2] was required.

The Scrum process is essentially an evolutionary, incremental framework. It is a team based software development approach, which uses a time-boxed adaptive artifact termed a Sprint. According to Millett et al. [13], Scrum takes an "iterative approach to software development." The Scrum process for software development is depicted graphically in Figure 1. It shows the key elements of the Scrum methodology much of which is referenced in this section.

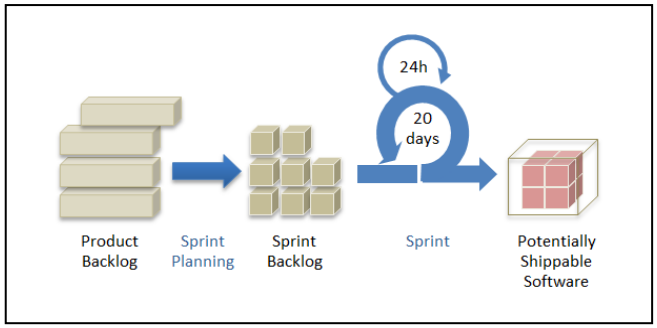


Figure 1. The Scrum Process

In Scrum, the team is responsive to its environment throughout the development. The developers are accorded unlimited flexibility and creativity during the development iterations. Knowledge is transferred among the team during the development and Schwaber [2] estimated that the probability of success using this approach would be high.

In accordance with its origins (the term Scrum comes from the sport of Rugby in which a group of players work together to move the ball up the field and over the try line to score points and win the game from the opposing team) it should be clarified that in addition to the various activities of the Scrum development process there are three roles, which help the team achieve success.

The Scrum Master is responsible for the team process, helping the team to achieve success and use Scrum correctly.

The team consists of cross-functional developers who between them possess all off the expertise necessary to deliver a potentially shippable increment of the product.

The Product Owner is a key role in Scrum as it is this individual who supplies the requirements that will comprise the product or system. Effectively a Scrum team works for a fixed duration (known as a Sprint) on product requirements. which are initially "contained in an ordered list known as the Product Backlog" [2]. At the beginning of each Sprint, the requirements are prioritized into a list known as the Sprint Backlog with the aim of completing an agreed set of deliverables by the end of the Sprint. Deemer et al. [14], explain further, "During the Sprint, the chosen items do not change. Every day the team gathers briefly to inspect its progress, and adjust the next steps needed to complete the work remaining. At the end of the Sprint, the team reviews the Sprint with stakeholders, and demonstrates what it has built. The development team obtains valuable feedback that can be incorporated in the next Sprint. Scrum emphasizes working product at the end of the Sprint that is really "done"; in the case of software, this means code that is integrated, fully tested and potentially shippable."

To add to this brief précis of Scrum this paper will now focus on some of the specific aspects that relate to the implementation of Agile Scrum. Barari [15] advises that "it is important to follow the guidelines defined in Scrum but the ultimate goal is to deliver what you promised." With regard to the guidelines, Schatz & Abdelschafi [16] state quite categorically that "there aren't many rules in Scrum but you need to adhere to the ones that (do) exist." These will be examined in the next section.

1) ASPECTS OF SCRUM SOFTWARE DEVELOPMENT

As mentioned above the core precepts for the implementation of Scrum are taken from Schwaber & Sutherland ([2][11]). It may be said that "the rules of transitioning software development from a plan-driven approach to an Agile approach are not set in stone" [1]. This is largely based on the fact that due to perceptual filters no two individuals will likely have the same interpretation of an agreed principle. However, there is much commonality attached to the writings on Scrum [11][17][18][19]; consequently, the next section will examine the activities of the Scrum development process as it is addressed by the authors listed above.

a) PRODUCT OWNER

There is complete unanimity on the requirement for the Product Owner role in Scrum. As described above the Product Owner has a key responsibility in the development process to supply the product requirements. According to Deemer et al. [14], "The Product Owner is responsible for maximizing return on investment (ROI) by identifying product features, translating these into a prioritized list, deciding which should be at the top of the list for the next Sprint, and continually re-prioritizing and refining the list. The Product Owner has profit and loss responsibility for the product, assuming it is a commercial product. In the case of an internal application, the Product Owner is not responsible for ROI in the sense of a commercial product (that will generate revenue), but they are still responsible for maximizing ROI in the sense of choosing – in each Sprint – the highest-business-value lowest-cost items."

With traditional methodologies the role of the developer was to elicit the Customers' needs in the form of the Requirements document. It was assumed, firstly, that the Customer knew exactly what he/she wanted and secondly, the Customer was able to document his/her needs in sufficient detail that the developer(s) would be clear on what was required. In contrast, Rico et al. [20] suggest that Agile advocates "listening to and interacting with Customers to ascertain their needs." This changes the requirements gathering from that of elicitation to a dialogue in which the role of the Product Owner is to interact with the Customer, to effectively be client-facing. Beyer [21] sees the Product Owner as "the Customer representative" and outlines his responsibility to "find out what the stakeholders and end

users actually need" [21]. According to Schwaber [2], the product Owner is "responsible for representing the interests of everyone with a stake in the project and its resulting system." Stober & Hansmann [18] define a Product owner who "represents the stakeholders, such as Customers." The Product Owner, then, effectively represents what Hauser & Clausing [22] refer to as the "Voice of the Customer." Pichler [19] states, "The Product owner must develop an intimate understanding of Customer and user needs, and how these needs can best be met." He suggests that the process for achieving this "is to involve Customers and users early and continuously in the development process" [19]. Furthermore, Pichler [19] recommends "asking Customers to provide feedback on prototypes, inviting Customer representatives to sprint review meetings and releasing software early and frequently are great ways to learn from Customers." Royce [10] also advises "it is important to involve the Customer in a formal way so that he has committed himself at earlier points before final delivery."

Whilst this is all encouraging there appears to not be a prescribed methodology to further these aspirations. Consequently, it is left to the software development organisation to establish the link between the Product Owner and the Customer. It would, however, appear that the requirements elicitation activity performed by the Product Owner is crucial to successful software development with Scrum.

In addition to the Product Owner, Scrum is very clear on the importance of involving the Customer. This is because lack of lack of user involvement is a primary cause of project failure. The CHAOS report of 2010 [23] stated: "projects that lack user involvement perform poorly."

b) CUSTOMER INVOLVEMENT

According to the published history of the Agile manifesto [6], the assembled group of developers espoused "a set of compatible values, a set of values based on trust and respect for each other and promoting organisational models based on people, collaboration, and building the types of organisational communities in which we would want to work" [6].

The proponents of Scrum, including Cobb [24], advocate "as much Customer collaboration as possible" but he counsels that the "Product Owner represents the voice of the Customer and is expected to provide overall direction to guide the project toward producing the value to satisfy Customer needs" [24]. This should most likely involve close collaboration with Customers and stakeholders.

Rico et al. [20] answer the question "How is Customer collaboration performed in agile methods?" They suggest the answer is "With right-sized, just-enough, and just-in-time interaction"[20]. With Scrum, Customer needs are captured in the form of epics and user-stories which form features within a product backlog. After some features have been implemented, Customer collaboration takes place after approximately 30 days in what is known as a Sprint review.

It would appear obvious that Customer involvement is crucial. Thus it is apposite to state that Customer involvement is an essential prerequisite to a successful software development process. However, it should be noted that the Product Owner and the Customer are but a part of the Scrum methodology. Without a team of cross-functional individuals to realize the Customer's vision the project would not get off the ground.

c) TEAM ORGANISATION

Traditionally teams were formed by managers in the organisation who appointed team leads and assigned personnel to the various roles that were required to develop a software product or system. Often teams failed to get results due to inappropriate team leadership, interpersonal dynamics or unclear objectives.

The Agile approach, according to Cooke [25] is to "rely on the mutual trust (and dependency) that emerges between stakeholders and delivery team members: delivery teams depend upon the expertise of stakeholders to accurately communicate and prioritize the business requirements; and stakeholders equally depend upon the expertise of the delivery team members to regularly produce outcomes that meet these requirements." It is this co-dependency between the team members and also between the team and its' Customer that makes Agile so powerful.

In order to achieve success, the Agile team must trust and depend on its members to perform their tasks to the best of their ability. Having this self-reliant and inward focus is undoubtedly a core strength of an Agile team.

Cooke [25] explains further, "Stakeholders are responsible for guiding the business priorities and for measuring the outcomes of each iteration, but they are not the people who determine the volume of work that can be achieved in that short time-frame. Instead, stakeholders defer to the multi-skilled delivery team to advise them on the actual work required to achieve their objectives, the estimated time for each task and what the delivery team can realistically achieve in an iteration given their current workload and other commitments."

Tata and Prasad [26] would appear to reinforce the need for self-organizing teams by explaining that "Self-management can indirectly increase team effectiveness by increasing team members' sense of responsibility and ownership of work."

d) SOFTWARE RELEASE

An important activity that is frequently mentioned by authors on Scrum is the activity of releasing software to Customers often and early. This is what Koch [27] refers to as "continuously and often". He cites the rationale behind this as to "increase (the) motivation for all participants, allow for easier discussion of the current status and therefore increase chances to uncover necessary changes and efficient possibilities for incorporating them."

Pichler [19] concurs, recommending "asking Customers to provide feedback on prototypes, inviting Customer

representatives to sprint review meetings and releasing software early and frequently are great ways to learn from Customers."

Thus, it would appear that there is evidence for this precept to be included in order to achieve a successful software development using Scrum.

e) SETTING TEAM PRIORITIES

According to Rawsthorne [28], the Product Owner (he refers to the simplest version of the Product Owner role as the Business Owner) "sits between the Stakeholders and the team." He cites the function of the Product Owner as being to "prioritize/order the work that the stakeholders want into a single Value Backlog." Furthermore, he explains the Product Owner "moves the items from the Stakeholder's Value Backlog to the Teams Work Backlog at a rate that will not overload the team" [28]. Khalil et al. [29], state, "in Scrum a single person must have final authority representing the Customer's interest in backlog prioritization and requirements questions. This person must be available (to the team) at any time to during the Sprint planning meeting and the Sprint review meeting." The setting of the team's priorities is a crucial part of the activity of the Scrum software development process.

Having outlined the guiding principles in Scrum the same approach will be taken with Extreme Programming (XP).

B. EXTREME PROGRAMMING (XP)

Extreme Programming (XP) - was introduced in 1999 by Kent Beck [3]. Rico et al. [20], describe that in its earliest incarnation XP featured "just-in-time evolution, self-chosen tasks, aggressiveness, model-driven development, and communications." Over the last decade, however it has evolved to incorporate practices which include "onsite Customers, pair programming, test-driven development, and open workspaces." Williams & Kessler [30] explain that "XP is a minimalist approach so it is essential that many of the practices actually get done." With that in mind, the fundamental aspects of XP (as depicted in Figure 2) will now be examined.

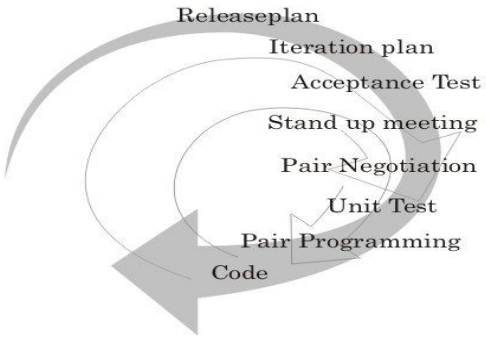


Figure 2. Planning and feedback loops in XP

2) ASPECTS OF XP SOFTWARE DEVELOPMENT

Both Scrum and XP are iterative albeit that the Sprint in Scrum would typically be slightly longer; a typical Sprint may last from two to four weeks whereas generally the iterations in XP would be from one to two weeks. In the same way as Scrum has its own defining aspects that are clearly identifiable as Scrum viz. Daily Scrum, Sprint Planning, Sprint Review, etc. so too, does XP have its own precepts. However, Kniberg & Skarin [31], state that "Scrum is less prescriptive than XP."

In XP, the role of the Product Owner is assumed by the Customer. The concept of a self-organizing team is still valid but whereas in Scrum there are no prescribed engineering practices, in XP there are some definitive requirements. These fundamental aspects of XP software development activities will now be examined in more detail.

a) CUSTOMER ON-SITE

Within Extreme Programming, Customer needs are captured in the form of user stories. The Customer is actually a full-time member of the project and communicates with the developers throughout the project." Cooke [25] concurs, "The most effective way to ensure ongoing business value is to *directly involve* key internal and external stakeholders in the process. In theory, representative stakeholders participate as *active members* of the Agile team during the process, providing the team with real-time input and hands-on feedback at two key points in the process:

- At the start of each iteration to describe and prioritise their business requirements
- At the end of each iteration to review and assess outputs against their stated requirements."

Ideally, stakeholders need to be on hand or perhaps, more importantly, they need to be available to the team in order to respond to developers' questions and review work as it is being completed. Cooke [25] opines, "The more available stakeholders are to the Agile team throughout the process, the closer that each deliverable will be to meeting the true needs of the organisation."

In XP, the Customer is actually part of the development team. When analyzing the role of the Customer it is interesting to refer to Beck's [3] intention that a "real Customer must sit with the team, available to answer questions, resolve disputes, and set small-scale priorities" … "someone who will really use the system when it is in production." Beck [3] would seem to advocate the presence of the Customer as a form of immediate feedback to the developer effectively being, according to Martin et al. [32] "someone who steps up and takes responsibility for the requirements." Furthermore, Martin et al. [32] argue that in XP the Customers "are charged with delivering what every developer wants: clear requirements, declared outcomes, and a helping hand with the messy world outside. XP makes development simpler by assigning some of the most slippery tasks to the Customer."

Thus, the Customer is critical to the success of delivering Agile software using the Extreme Programming (XP) methodology.

Stober & Hansmann [18] would appear to concur, stating "the project team always needs to keep in close contact with the Customer to ensure that the project is meeting the Customer's expectations at any given time."

b) PAIR PROGRAMMING

Cockburn [4] classifies XP as a "high discipline" process since there are various prescribed practices outlined.

One of the key activities most often associated with XP is that of Pair Programming. This is where two developers work together at one workstation. Williams & Kessler [30] describe pair programming as "an integral part of XP."

Quite apart from the obvious benefit of two problem solvers working on the same issue and discussing optimal solutions there is the added bonus of continuous code review. Williams & Kessler [30] caution that "it is dangerous to do XP without pair programming."

c) SELF-ORGANIZING TEAM

Shore [33] explains that XP teams are self-organizing and cross-functional. Moe et al. [34] use the label "self-organizing" teams as a synonym for "autonomous teams" and for "empowered teams". They refer to the work of Guzzo & Dickson [35] and explain such teams as "teams of employees who typically perform highly related or interdependent jobs, who are identified and identifiable as a social unit in an organisation, and who are given significant authority and responsibility for many aspects of their work, such as planning, scheduling, assigning tasks to members, and making decisions with economic consequences."

d) SETTING PRIORITIES

In XP it is the Customer who prioritizes the work to be done. Griffin [36] explains, "Customer ability to determine what developers will work on next greatly increases the Customer's sense of confidence in their development unit, and sense of being listened to. The Customers began to feel for the first time that they were truly in control, and that they would receive what they needed when they needed it."

e) OPEN PLAN WORKSPACE

Robinson & Sharp [37] explain that an open plan workspace, which is often synonymous with XP software development is "symbolic of the culture. The physical setting is open plan: open and public to all in the team." The obvious rationale behind this practice is that it fosters good communication between the team. Recht & Nielson [38] state, "Communication is the basis of XP. Any problem occurring can invariably be traced back to lack of communication either between the developers or between the developers and the Customer. As such, it is important never to let communication become a secondary priority."

Having examined both Scrum and XP and highlighted their various key activities, practices & roles and the rationale behind them the next stage is to describe the research that was conducted to ascertain the level of compliance to these clear and unambiguous Agile precepts.

The next section presents the research method adopted in addition to the research vehicles and participants used in the study.

IV. THE RESEARCH

Countless academic papers, textbooks and instruction manuals have been written describing the various Agile approaches/techniques but, to date, to the best of this author's knowledge, with the exception of research conducted by Salo & Abrahamsson[12] on the use and usefulness of XP and Scrum in European embedded software development organisations no one has sought to identify whether the existing paradigms, in their implementation, do, in actuality, follow the precepts as laid down by the proponents of these self-same methodologies.

Specifically, this refers to whether software development organisations who self-report as using either Scrum or XP are really implementing the method as intended or indeed whether (as is hypothesized by the author) the organisations are adopting an *ad hoc* approach to implementation, a 'pick-and-mix' approach, in effect, where those precepts that suit the organisation are adopted and those that would require a fundamental shift in the organisation's culture are sidelined.

Basically, the author wished to understand the level to which management/developers follow or are allowed to follow the process. An integral part of the research was to identify whether those managers and developers who admit to working in Agile software development organisations actually believed their organisations to be Agile (the view from the coal-face, one might say).

Consequently, the purpose of the research was to firstly clarify which methods were being used for software development and then to identify the perceptions of those that worked in these self-reported Agile organisations. In addition, the author wished to establish empirically whether the precepts of the practiced Agile methodology were being inculcated into the actual development process as has been advocated by the pioneers of the methods.

It was believed that the answers to these research questions would add significantly to the body of knowledge regarding Agile implementation in Irish software development organisations.

A. Research Method

The research effort in this case was centered on conducting a quantitative study that would be descriptive in nature. Leedy & Ormrod [39] describe this type of research as "identifying the characteristics or exploring possible correlations among two or more phenomena." The authors also state "descriptive research examines a situation as it is" [35].

The chosen method that was used was that of the survey, which, according to Leedy & Ormrod [39], "involves acquiring information about one or more groups of people by asking them questions and tabulating their answers" Leedy & Ormrod [39] indicate that "the ultimate goal is to learn about a large population by surveying a sample of that population." Due to time constraints and logistics it was decided to use an online survey. However, with a view to gaining a deeper insight interviews were also conducted when it was felt a response required clarification or more detail was sought.

In an ideal scenario, it would have been preferable to obtain a totally random selection of employees in Irish software development companies to answer the research questions. However, there was a concern that if the response rate was low (which is one of the main drawbacks of this research method, what Leedy & Ormrod [39] refer to as "low return rate") then the research may have been over before it began. Consequently, it was decided to adopt a degree of 'selective sampling'. This is what Nardi [40] refers to as "purposive sampling." This involved including specific pre-defined groups in the sampling frame with a view to increasing the likelihood of collecting "data on organisations that had some prior knowledge of Agile practices, as opposed to taking a completely random sample, which may have resulted in confused responses. [1]"

In addition to personal contacts it was decided to 'acquire' a list of /access to software development companies from software groups such as AgileIreland, Information Technology Association Galway (ITAG), the Irish Software Association (ISA), the Irish Software Innovation Network (ISIN) training companies, blogs etc. and also from colleagues/past students of the NUI Galway MScSED course who have contact with the software development industry. A very brief description of the various software support groups is listed in Section IV. C, which deals with the survey participants.

B. Research vehicles

Whilst the primary focus of the research was on organisations that use either Scrum or XP it was decided to use the research to gather as much data as possible.

With this in mind, it was decided to stratify the research into two focal domains. First there was the Agile management domain which would be comprised of those involved in the management of Scrum/XP development projects. Specifically roles such as Software Development Manager, Project Manager, Test Manager, Scrum Master, Release Manager etc. were expected to feature in this category.

The second domain was that of those who were more 'hands-on' in the software development process. It was hoped to get information from those who would be involved in the software development team.

In order to make the research manageable it was decided to create two surveys which would be made available to the participants who would then be free to choose the one most applicable to their role in software development. Clear and unambiguous instructions accompanied links to the surveys both informing would-be participants of the research and also guiding them to choose the most appropriate vehicle for them to access. Having identified the research questions the survey was created by the author and prior to release the survey was validated by a cross-functional set of academics and software industry representatives. When it was felt that the surveys as developed were capable of generating clear, unambiguous data they were hosted online by SurveyMonkey.

For ease of reference the first grouping was labeled the Management survey and the second domain was referred to as the Developer/Team survey.

To a large extent both surveys contained many of the same questions although the actual detail of the team working practices were omitted from the Management survey. The next section looks at the research participants.

C. Participants

As previously stated purposive sampling was used to target would-be participants due to the requirement to ensure that the data obtained would come from organisations that were conversant with Agile software development. Research into the Irish software development community suggested the following be included:

1) AgileIreland

AgileIreland is an Irish web community. According to its mission statement AgileIreland is "a community site for anyone interested in agile and lean methods of software development throughout the 32 counties of Ireland." Unfortunately, there is no information on the site about the number of members who might potentially access the surveys. The AgileIreland discussion boards were used to launch the online survey on their site.

2) Information Technology Association Galway (ITAG)

A large number of local software development companies are members of ITAG, which is the Information Technology Association, Galway. According to their web Home page ITAG "was established in 2000 by a group of forward looking IT professionals representing both multinational and indigenous IT companies. Our goal is to foster the continued growth of a strong IT cluster in the Galway region, through networking and training events, joint initiatives, and regional and national advocacy." The author contacted ITAG with a view to having the online survey link circulated within the software development industry in the West of Ireland.

3) The Irish Software Association (ISA)

The Irish Software Association (ISA) is a part of IBEC (the Irish Business & Employers' Confederation). Their web presence states that their "membership base is comprised of over 160 companies actively involved in every area of the software sector in Ireland." They have a strong Internet base and host the ISA LinkedIn pages. The LinkedIn membership

is quoted at 1,416 and it was to this potential audience that the author publicized the online survey.

Obviously, with annual vacations etc., it was not realistic that the author's online survey would be accessed by the entire membership, but the author was hopeful that a fraction would be sufficiently cooperative to acquiesce to completing the surveys.

4) Personal Contacts/Tutors/Past MScSED students

It was likely that much of the data collected from the survey would be obtained from the author's personal contacts, as it is an established fact that people with whom one has an existing connection are likely to be more responsive to requests for information.

Based on all of the various channels described above the author was confident that little more could be done to solicit responses to the research questions. However, whilst the potential audience for the research was estimated to be in the region of 2,000 individuals the author was well aware that only a small fraction would take the time and trouble to respond.

With a view to capturing a fully representative view, cross-functional participants, including both Agile team members and software development management in organisations that self-reported as using Scrum and XP, were targeted. In this way it was hoped that the findings would be representative of the actual state of play of software development in Irish industry.

The complete breakdown of Scrum management participants is shown in Table I.

TABLE I. MANAGEMENT SURVEY PARTICIPANTS

	Organisation Size		
Role	***1 to 50***	***51 to 500***	***500+***
S/W Dev. Mgr.	3	3	4
Project Mgr.	2	2	4
Q.A. Mgr.			1
Test Mgr.		1	

Similarly, the breakdown of Scrum team participants is shown in Table II.

TABLE II. SCRUM TEAM SURVEY PARTICIPANTS

	Organisation Size		
Role	***1 to 50***	***51 to 500***	***500+***
Designer	1	1	1
Senior Developer	2	3	1
Developer	2	4	3
Test Engineer	2	3	2

Given the fact that the survey was online, it was not possible to compute a response rate, *per se*. However, given the purposive sampling involved and the profile of a number of the organisations represented (e.g., Cisco, Globoforce, Avaya, CSG International, etc.) it was felt that a sufficiently representative number of respondents to the Scrum questions had contributed to make the results relevant.

Unfortunately, the level of XP response was extremely disappointing. In total, only three responses who reported using XP were received. However, the data that was collected will, nevertheless, be included though one would have to refrain from making any pronouncement based on limited statistical data.

V. RESULTS

As explained previously, at the commencement of the research activity the author sought to classify the respondents to enable a proper context to be applied to the research questions. Participants had access to both a Management and a Developer/Team survey. Clear instructions were provided and the respondents were then invited to select the survey that best described their role in the Agile organisation within which they worked.

A. Management Results

In order to provide a degree of clarity this section commences with an introduction to the profile of the respondents surveyed. This includes organisational size and sector in addition to the respondent's role in the organisation. The subsequent section presents the Scrum results and finally the XP results are presented.

1) Respondent Characteristics

It can be stated that those who completed the Management targeted survey encompassed organisations of varying size, i.e., 28% from organisations with fewer than 100 software personnel, 41% from organisations with between 101 and 500 personnel involved in software development and 31% coming from organisations with more than 501 personnel involved in software development activities. All organisations were based in Ireland with a preponderance coming from the west coast of Ireland. This is shown graphically in Figure 3.

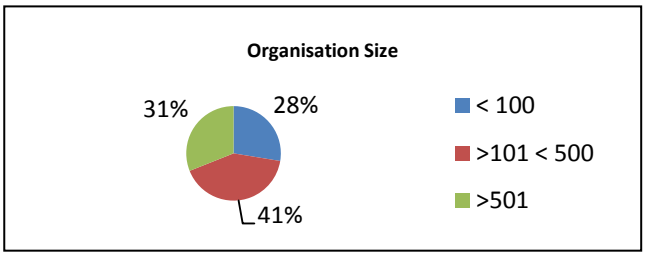


Figure 3. Organisation size of management survey respondents.

The next classifier was the nature of the respondents' organisation. As part of the survey validation process it was decided to include the organisation sector to see if there

were any discernible patterns in the data. The overwhelming category represented is in the sector of telecommunications/unified communications with a 55% representation. 21% represents software products ranging from reward & recognition software to security. The overall breakdown is shown in Figure 4, below.

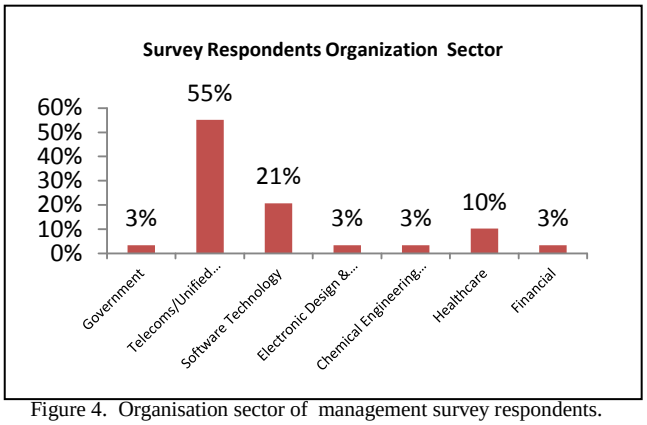


Figure 4. Organisation sector of management survey respondents.

The final part of the respondents' classification focused on their role within the software development process. Note that of the two surveys available to respondents this particular survey was targeted only at the organisational /managerial level of the software development activity. Consequently, the author would expect to only see responses from personnel involved in the management and deployment of the software development activity in the organisation in which they operate. In a software development organisation this would most likely encompass a broad spectrum from Software Development Manager to Project Manager, Scrum Manager, QA Manager, Test Manager, etc. as shown in Figure 5.

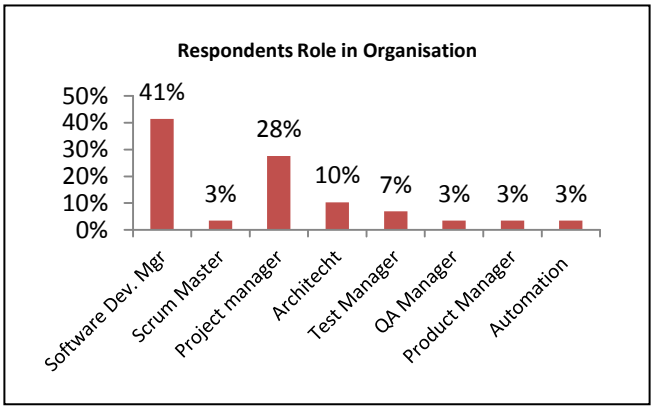


Figure 5. Management survey respondents roles in organisation.

The range of roles selected by the respondents validates that the author did, indeed, capture views of those individuals involved in the management of the organisations' software development activities.

In terms of the type of Agile method that predominates in the organisations that admit to having an Agile software development process the overwhelming method used is Scrum with 83% of all respondents selecting it. XP features in a minor sense (17%) and other techniques such as Crystal Clear do not feature whatsoever in this survey although they were offered in the survey as a possible option. The results are displayed in Figure 6.

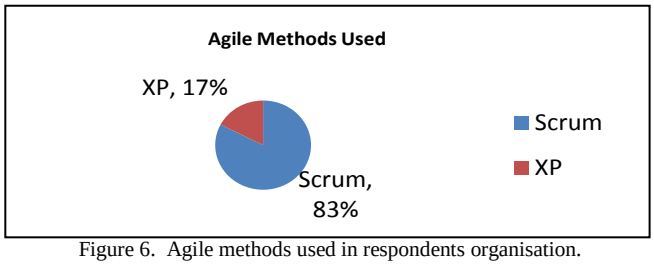


Figure 6. Agile methods used in respondents organisation.

a) Scrum Management Respondents Results

Given the self-described adoption of Agile one might expect that the survey respondents would perceive their organisation to, in fact, be operating in a fully Agile manner. This formed the basis for the next question which asked whether the respondent considered his/her organisation to be truly agile in the context of handling and managing software churn. The results are presented in Figure 7.

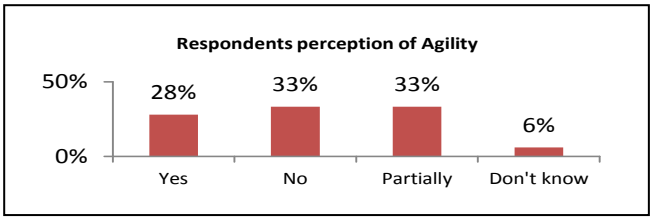


Figure 7. Management respondents perception of organisation's agility.

A number of respondents confirmed these views in subsequent interviews admitting that whilst the perception by many (senior management, Customers, etc.) is that they are committed to Agile, the organisation for which they work is, in fact, only paying "lip service" (Respondent #17) to being Agile. There appears to be a degree of frustration that respondents are not being enabled to implement the techniques correctly and this is due in part to pressure to "keep the product gates open."(Respondent #4)

In addition to assessing agility, it was felt to be important to determine if survey respondents were satisfied with their existing Agile software development process. The overall answer to this question from the Management perspective is displayed in Figure 8.

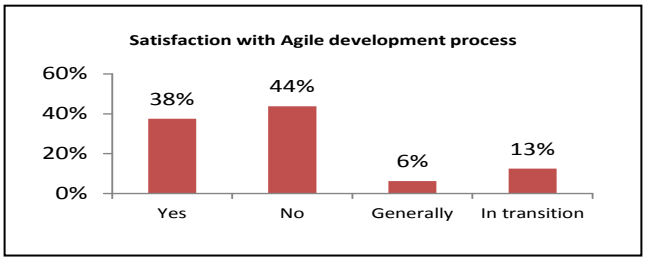


Figure 8. Management survey respondents satisfaction with Agile process.

To conclude this section the author sought to clarify the level of Customer involvement in the development process as perceived by the respondent who is in an Agile management role as opposed to an operational role. The question asked whether the respondent's organisation actively encouraged Customer involvement in the software development process. The answer, displayed below as Figure 9 was an overwhelming Yes at 86%.

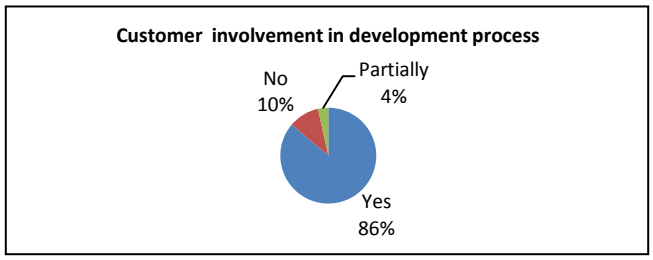


Figure 9. Management respondents perception of customer involvement in software development process.

Of those that responded No it is interesting to note that they represent organisations who claim (self-describe) to be Agile. The respondent who acknowledged that Customers were "partially encouraged" (Respondent #8) was also employed in a self-described Agile organisation.

With a view to affirming the actuality of Customer involvement the respondents were asked whether software was released/demonstrated early to Customers. It would be logical to expect that Customer involvement and attendance at Sprints to see working software would yield similar results. The responses are displayed in Figure 10 below.

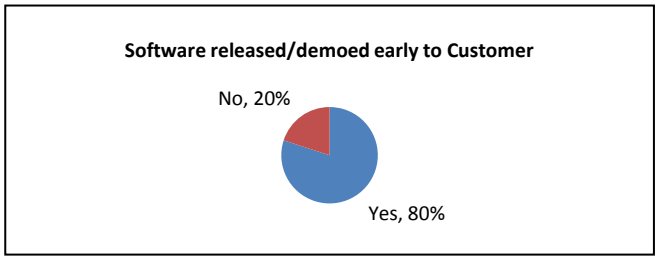


Figure 10. Management respondents knowledge of early software release.

The final question related to how management believed their Customers perceived their organisations' Agile development process.. Somewhat surprisingly there was not an overwhelming majority as can be seen in Figure 11.

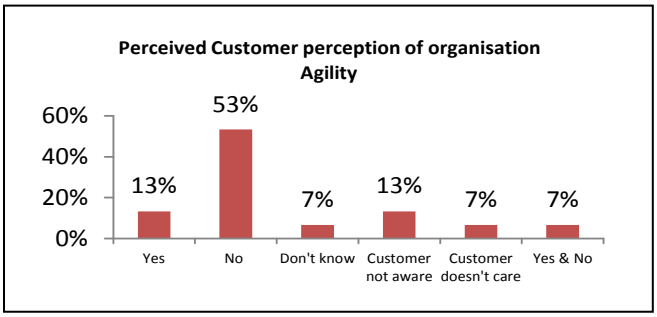


Figure 11. Management respondents view of customer perception of organisational Agility.

B. Developer/Team Results

In light of the management perception of the software development process portrayed in Section V.A the author hoped that the views of the development team might shed further insights into the software development process. More than half of the surveys collected (53%) represented the views of developers to many of the same questions but in addition the respondents were asked about the details of their software development processes.

As with the Management survey, initial questions focused on obtaining a profile of the respondents.

1) Respondent Characteristics

In terms of the size of the organisations in which the respondents work the analysis as shown in Figure 12, describes 43% as working in companies of less than 100 people, 33% between 101 and 500 and 24% in large organisations of greater than 500 people.

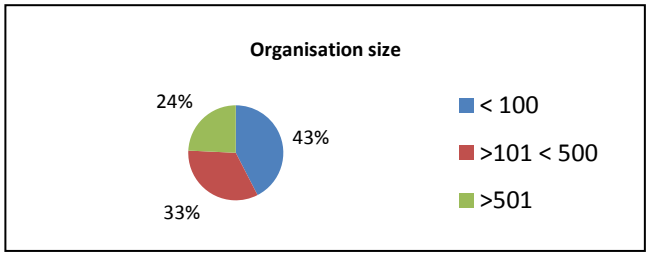


Figure 12. Developer survey organisation size.

As with the Management survey the respondents were representative of a diverse range of business sectors although as before, the telecoms/unified communications sector dominated at 46%. The software technology sector, as before, encompassed security and reward recognition software in addition to some unspecified software products. This is displayed below as Figure 13.

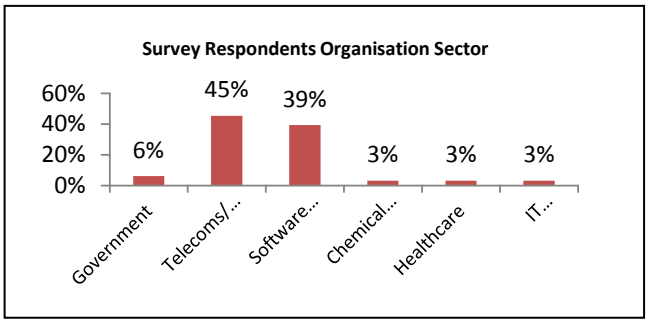


Figure 13. Developer survey respondents organisation sector.

Finally, the role of the respondents in their respective organisations was categorized as shown in Figure 14. It must be noted that the Part 2 survey was targeted exclusively at non-management personnel. It was hoped that the respondents profile would be that of the 'worker at the coalface' of software development (a software development team member) thus enabling the author to obtain the views of those individuals actively involved in

the software development process. As shown in Figure 14 the survey did, indeed, capture a good cross section of these individuals.

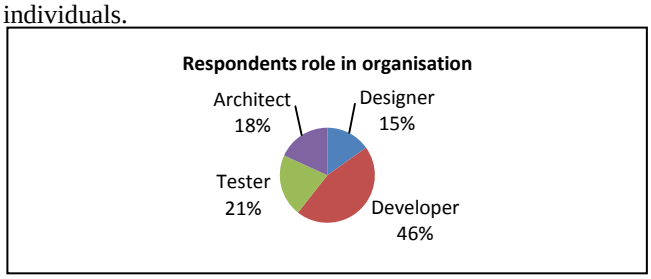


Figure 14. Developer survey respondents roles in organisation.

As before, when questioned about the Agile method used in their organisation to develop software the results were overwhelmingly in favour of Scrum. This is shown in Figure 15 below.

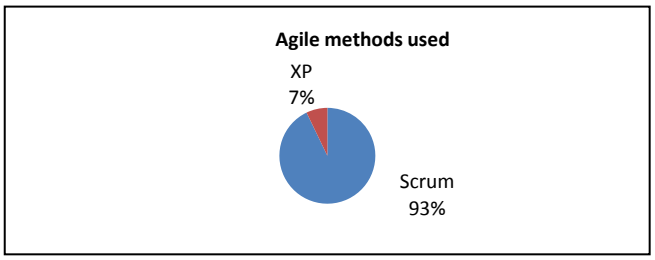


Figure 15. Agile methods used in respondents organisation.

a) Scrum Team Respondents Results

In light of those who stated that their organisation was Agile the author sought to verify that, in the context of their role, the survey respondents had the perception that their organisation was truly Agile when dealing with software churn. Figure 16 presents the overall picture.

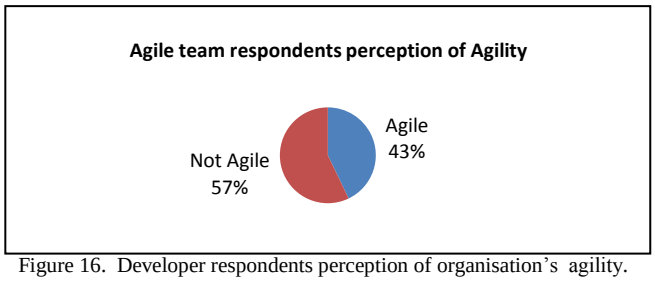


Figure 16. Developer respondents perception of organisation's agility.

As with the Management section of the survey the author wished to ascertain the level of satisfaction with the current Agile process being used. This is shown as Figure 17 below.

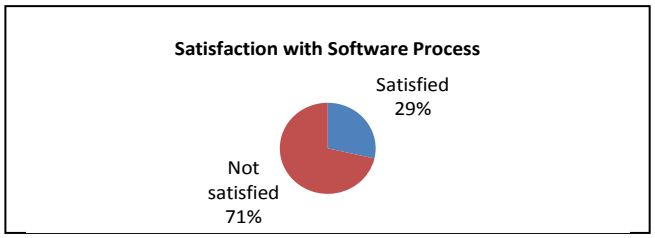


Figure 17. Developer survey respondents satisfaction with Agile process.

b) Scrum Team Process Results

The data collected in the next section specifically answers the questions on the implementation of those key aspects of the Scrum methodology that were detailed in Section III of this paper.

Of those organisations that use Scrum, Figure 18 displays the breakdown of perceptions regarding Customer involvement in the software development process.

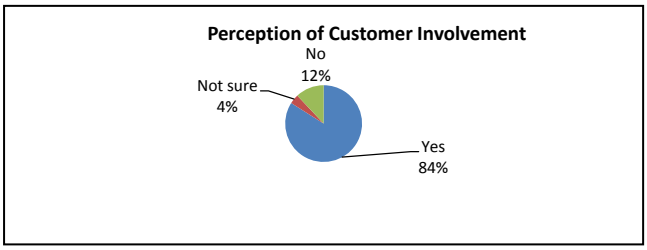


Figure 18. Developer respondents perception of Customer involvement in software development process.

100% of respondents admitted that the role of Product Owner was actively incorporated in their Scrum team.

This was followed by asking how often the Product Owner consulted with the Customer. The results are displayed in Figure 19 below:

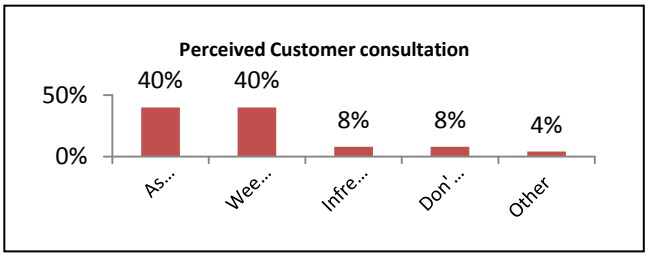


Figure 19. Team respondents perception of Customer consultation.

With regard to whether or not the software development process in the respondents' organisation facilitated the use of self-organised teams the respondents surveyed returned the breakdown as displayed in Figure 20.

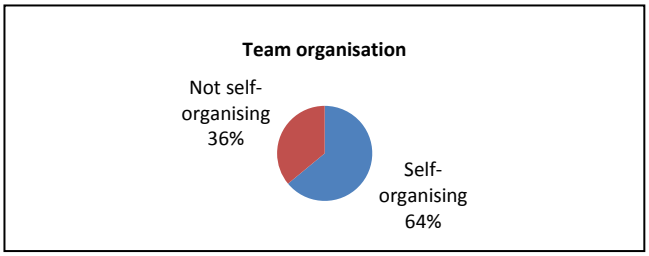


Figure 20. Developer survey respondents team organisation status.

One of the key tenets of Scrum is that working software should be released regularly and/or demonstrated to Customers to solicit feedback. Respondents were asked if software was released early to Customers according to this principle. The results are shown in Figure 21 below.

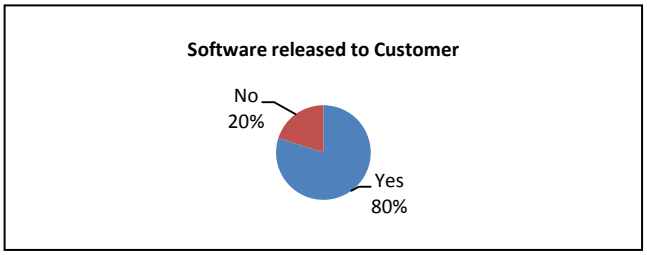


Figure 21. Software released for early feedback.

The final but nevertheless important question for Scrum teams centers around who sets the priorities for the team. As can be seen from the pie-chart in Figure 22 the results are somewhat varied.

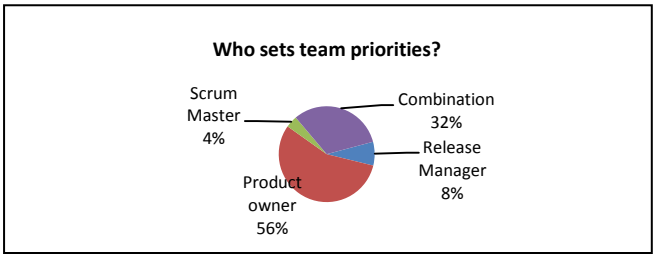


Figure 22. Setting of team priorities.

c) XP Respondents Results

The responses to the XP questions are not statistically valid given that the sample size was only three responses. However, in the interest of completeness, the results will be presented.

The first question to be asked of those respondents who described their software development process as Agile using XP concerned whether the Customer (or their designated, capable representative) was on-site during the development process. The results of this are displayed in Figure 23 below:

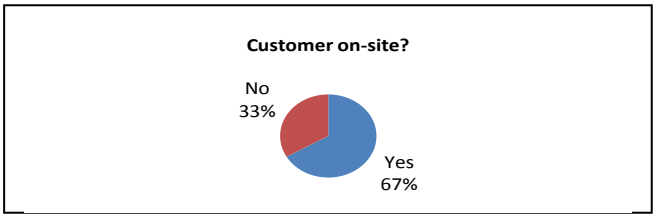


Figure 23. Presence of on-site Customer.

In response to the question on open-plan seating it transpired that all respondents claimed the team was situated in an open workspace.

Next, the issue of pair-programming was examined. The results are shown in Figure 24.

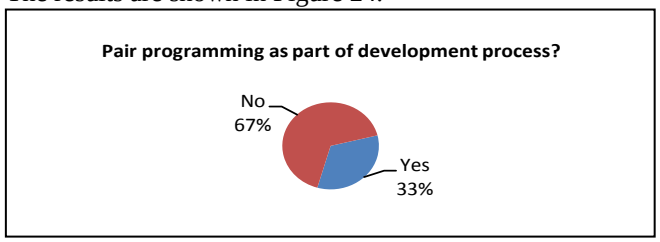


Figure 24. Utilization of pair programming.

Similar results were obtained for the team organisation in that as shown in Figure 25, 66% of respondents described their XP team as non-self-organizing.

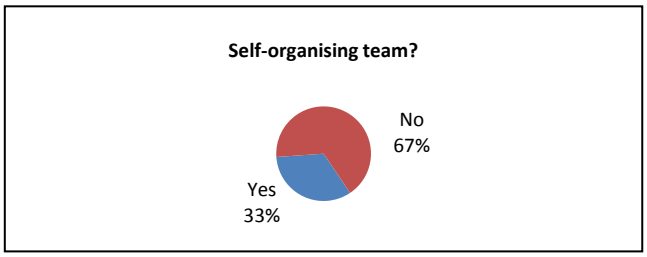


Figure 25. Utilization of self-organizing team.

The next section attempts to 'make sense' of the collected data with a view to answering the initial hypotheses and adding some general observations about the research process itself.

VI. DATA ANALYSIS/FINDINGS

As previously stated, the purpose of this research was to initially clarify which methods were being used for Irish software development. Following from this the author surveyed software professionals at both management and developer level in Irish companies to examine the perception of the agility of the software development processes being used and also to identify whether the precepts of the chosen Agile method were being inculcated into the actual development process as has been advocated.

The Management survey encompassed a broad spectrum of software development process management with Software Development managers being the largest sector at 42%. However, other senior/management roles from Project manager, Test manager, Scrum manager and Architect were also represented. Consequently, the author believes that a sufficiently wide management perspective was attained.

It can be observed from Figure 7 that those that claim to be fully Agile do not always perceive their own development process in this light. In fact, only 27% of the Management surveyed felt that their process was, indeed, Agile, 33% answered partially Agile but worryingly 33% of these respondents did not perceive their process as Agile. This is quite a surprising but nevertheless important result.

Perhaps the previous statistic accounts for the lack of satisfaction reported by Management respondents with the existing Agile processes in which those surveyed operate. 44% of Management respondents (Figure 8) reported a lack of satisfaction with the way in which Agile was being implemented. In terms of Customer perception, the author acknowledges that this should really be a question asked of Customers of the organisations surveyed. However, due to the logistics involved the author opted, instead, to ask the managers in the software development organisations if they were of the opinion that their Customers perceive their software development process as Agile. Whilst this is clearly a subjective question it was nevertheless included in order to identify whether, if, as the

author contended an ad hoc approach to Agile software development was being used, the Customer was aware of this situation. Management were thought to be sufficiently close to such important project deliverables as on-time delivery, on-budget, and content as to be able to make a judgement call on their Customer's perceptions of the software development process. The findings of this question are that the majority of respondents (53%) do not. This is shown in the bar chart of Figure 11.

It was hoped that what is referred to, as the Developer/Team survey would represent the views and opinions of those individuals who are actually engaged in the 'hands on' activity of software development. This, indeed, proved to be the case as an analysis of the respondents' profiles showed that 46% were engaged in software development with test, design and development equally represented in the remaining 54%.

For the Developer survey it was interesting to note that 57% of those surveyed would not perceive their process as Agile. (cf. Figure16). One explanation for this listed by a respondent was "we are not yet fully Agile." Conceivably others who did not perceive their process as Agile had similar misgivings.

Possibly the most salient finding of the research dealt with the respondents' perception of the implementation of Agile precepts in their software development organisation. The author believes this to be one of the key elements of the research as it may explain the high level of dissatisfaction (44% according to Figure 8) reported by management with the Agile software development processes. Incidentally the level of dissatisfaction at the operational/developer level was significantly higher with 71% of respondents claiming to be dissatisfied with their software development process (cf. Figure 17).

One would have to have serious misgivings about the morale of these organisations. In any organisation were staff are felt to be dissatisfied with the way in which the development process is being conducted it would not be surprising to find a knock-on effect of software development ineffectiveness. The respondents surveyed were asked about the success or otherwise of projects they had been involved in in the past two years. The results of this are shown in Figure 26.

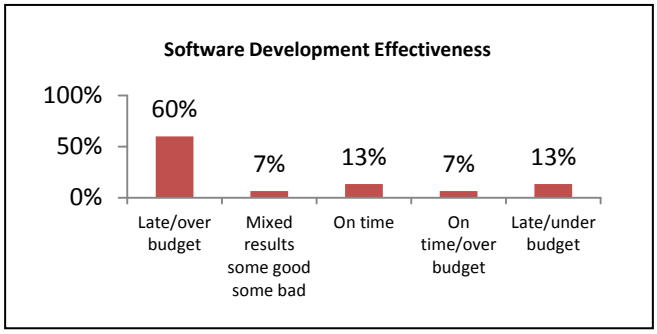


Figure 26. Software Development Effectiveness

With regard to the full implementation of Agile software development methods, specifically Scrum in those organisations represented by the respondents who completed surveys the author has used the data collected by the survey research to generate a table. The table, presented below as Table III, shows the adoption, or lack thereof, of the various key aspects advocated by the proponents of the methodology and explained in Section IIIA.

TABLE III. AGILE PRECEPTS ADOPTION

Suirvey #	Customer involved?	Who sets priorities?	Self org. team?	s/w rel early?
1	Yes	Release Mgr.	Yes	Yes
2	Yes	Release/Scrum	Yes	Yes
3	Yes	Product Owner	No	Yes
4	Yes	Product Owner	Yes	Yes
5	Yes	Product Owner	Yes	Yes
6	Yes	Release Mgr.	Yes	Yes
7	Yes	Prod. /Rel Mgr.	Yes	Yes
8	No	Product Owner	No	Yes
9	Yes	Product Owner	No	Yes
10	Yes	Product Owner	Yes	Yes
11	Yes	Product Owner	No	Yes
12	Yes	Product Owner	No	Yes
13	Yes	Product Owner	No	Yes
16	Yes	Product Owner	Yes	Yes
17	Yes	Product Owner	Yes	Yes
18	No	Mysterious pro	No	Yes
22	Yes	Scrum Mgr/Pro	Yes	Yes
23	Yes	Scrum Mgr/Pro	No	Yes
24	No	Scrum Mgr/Pro	Yes	Yes
25	Not sure	Scrum Mgr/Pro	Yes	Yes
26	Yes	Product Owner	Yes	Yes
27	Yes	Product Owner	Yes	Yes
28	Yes	Release/Scrum	No	Yes
30	Yes	Product Owner	Yes	Yes
33	Yes	Product Owner	Yes	Yes

As can be seen from Table III only 28% of the respondents are working in organisations that adhere to all of the Agile Scrum guidelines.

It should be noted that due to the limited amount of data collected this has not been done for the software development organisations that claimed to use XP.

VII. DISCUSSION

Firstly, this research confirms previous international findings that Scrum is the predominant Agile methodology in use in software development. In this sample of Irish software development industry this finding was found to hold true in both the Management and Team surveys with 83% of those in Managerial positions and 93% of Scrum team members reporting it as their organisations' Agile method of choice.

In terms of the actual research it was found in terms of the Scrum precepts, and, notwithstanding that the XP dataset was small, the XP precepts, that the actual implementation of the Agile methodologies was not as rigorous as had been hoped. Rather, the author's contention

that organisations adopt an *ad hoc* approach to implementing Agile has been borne out.

Table V demonstrates the argument for this contention. Out of 33 surveys from team members who self-described as using Scrum only 18% were following all of the guidelines of Scrum as described by its proponents. This discounts the possibility that a limited number of 'rogue' organisations were not using a number of the Scrum guidelines. Rather, the data shows that 82% of those organisations surveyed were falling short in at least one regard.

There are many possible reasons for this. Firstly it is conceivable that organisations who have recently transitioned from a traditional approach to software development are experiencing difficulty in 'letting go' of the formal chain of command that frequently accompanied the more traditional plan-based methodologies, e.g., Waterfall. This would account for 30% of the anomalies in setting team priorities.

This same rationale would also account for the non-self-organizing teams. In order to transition to an Agile environment often the organisational culture will have to be changed to facilitate autonomous teams who are responsible for achieving team goals and managing their own workload. This would account for another 27%.

Secondly, when it comes to Customer involvement, this is a difficult arena where it is necessary to foster a trusting partnership with the Customer. It can be truly daunting to open up a software development organisation to the Customer and expose the organization's internal workings.

Based on the research it would appear that organisations who lay claim to being Agile are taking on board those guidelines which are relatively easy to implement. A case in point is the appointment of a Product Owner. As the Product Owner is often referred to as "the single wringable neck [2]" it is relatively easy to change the Traditional model role of Project Manager into the Agile Product Owner.

This theory is also borne out in the, albeit, limited data available on XP in that all of the survey respondents acknowledged an open plan workspace, which requires little organisational commitment but eschewed Pair Programming, which would require a paradigm shift in the software development operation.

VIII. LIMITATIONS & FUTURE WORK

It needs be stressed that survey research "captures a fleeting moment in time" [41]. It is completely possible that the response to a particular question might be totally different in the future as circumstances alter. Once this was taken on board, however, it was felt that a survey would be a perfectly acceptable way to discover information about this research topic. De Vaus [42] states "Survey research is widely regarded as being inherently quantitative and positivistic and is contrasted to qualitative methods that involve participant observation, unstructured interviewing, case studies, focus groups, etc. Quantitative survey research is sometimes portrayed as being sterile and unimaginative but well suited to providing certain types of factual, descriptive information – the hard evidence."

"If survey research has a drawback it would seem to be that the results are dependent on the participants' willingness to participate in addition to their ability to correctly answer the questions asked" [1]. Leedy & Ormrod [39] refer to the fact that the method relies on "self-report" data. The authors caution that "people are telling us what they believe to be true or, perhaps, what they think we want to hear."

Perhaps the greatest limitation of this research is its relatively small sample size. In total, the survey respondents numbered 45 individuals (cf. Table I and Table II). The margin of error on such a small sample is 14% but the author believes that due to the combination of quantitative and qualitative techniques employed the results are nevertheless indicative of the actual state of the Agile software development processes in Ireland. It was hoped that more data could have been obtained but given the short timeframe – the research was effectively conducted during the Summer of 2011 as part of a Masters dissertation in Software Engineering (MScSED) – this proved not to be the case.

Future work in this domain is ongoing specifically in the realm of Agile Scrum teams.

IX. CONCLUSIONS

The goal of this research was to add to the existing body of knowledge regarding Agile implementation in a sample of Irish software development organisations.

Conboy [43] declares "there is no consensus as to what constitutes an agile method." Undoubtedly, this research would agree with that statement.

The research set out to ascertain whether Agile practices are being implemented rigorously. The results would seem to indicate that, as hypothesized, this is not the case.

One would wonder if the lack of satisfaction with the respondents Agile processes could, in part, result from such an *ad hoc* approach. As Addison & Vallabh [44] advocate to control software projects it is important to "develop and adhere to a software development plan."

As was explained in Section III of this paper there is good rationale underpinning all of the Agile precepts and consequently there needs to be a similar well-reasoned rationale for excluding these self-same guidelines.

REFERENCES

[1] T. O'Connell, "The Scrum Product Owner – Customer Collaboration & Prioritizing Requirements," in Proc. ICSEA13, Venice, Italy, pp. 368-372, 2013.

[2] K. Schwaber, "Scrum Development Process," in Proc. OOPSLA'95 Workshop on Business Object Design and Implementation, Austin, Texas, USA, pp. 117 -134, 1995.

[3] K.Beck, Extreme Programming Explained: Embracing Change, Boston, MA. Addison Wesley, 1999.

[4] A. Cockburn, Agile Software Development, Boston, MA. Addison Wesley, 2001.

[5] VersionOne "7th Annual State of Agile Development Survey," retrieved 2014.01.20 from http://www.versionone.com/state-of-agile-survey-results/

[6] M. Fowler and J. Highsmith, "The Agile Manifesto," Software Development, Vol. 9, No. 8, pp. 28-32, Aug. 2001.

[7] L. Buglione, "Light Maturity Models (LMM); an Agile Application," Proc. of the 12th International Conference on Product Focused Software Development and Process Improvement, pp. 56-57, 2011.

[8] S. Ambler and M. Lions, Disciplined Agile Delivery: A Practioner's Guide to Agile Software Delivery in the Enterprise, Boston, MA. Pearson Education, p. 2, 2012.

[9] J. Dooley, Software Development and Professional Practice, New York, NY: Springer, p. 29, 2011.

[10] W. Royce., "Managing the Development of Large Software Systems," Proceedings of IEEE WESCON Vol.26, No. 8, pp. 1-9, Aug. 1970.

[11] J. Sutherland, "Agile Principles and Values," MSDN, retrieved 2014.02.04 from http://msdn.microsoft.com/en-us/library/dd997578.aspx

[12] O. Salo and P. Abrahamsson, "Agile methods in European embedded software development organisations: a survey on the actual use and usefulness of Extreme Programming and Scrum," in Proc. IET Software, Vol. 2, Issue 1, pp. 58–64, 2008.

[13] S. Millett, J. Blankenship, and M. Bussa, Pro Agile. NET Development with SCRUM, New York, NY: Apress, p.13, 2011.

[14] P. Deemer, G. Benefield, C. Larman, and B. Vodde, "Scrum Primer v 1.2," Retrieved 2013.08.17 from http://goodagile.com/scrumprimer/scrumprimer.pdf.

[15] T. Barari, "Tips for First Time Scrum Masters," Scrum Alliance, 2009. Retrieved 2013.08.16 from http://www.scrumalliance.org/community/articles/2009/may/tips-for-first-time-scrummasters.

[16] B. Schatz and I. Abdelschafi, "Primavera Gets Agile: A Successful Transition to Agile Development," IEEE Software, Vol. 22, no. 3, pp. 36-42, May/June 2005.

[17] J. Highsmith, Adaptive Software Development Ecosystems, Boston, MA: Pearson Education Inc, pp. 244-245, 2002.

[18] T. Stober and U. Hansmann, Agile Software Development: Best Practices for Large Software Development Projects, Berlin Heidelberg: Springer-Verlag, 2010.

[19] R. Pichler, Agile Product Management with Scrum: Creating Products that Customers Love, Boston, MA: Pearson Education Inc, 2010.

[20] D. Rico, H. Sayani, & S. Sone, The Business Value of Agile Software, Fort Lauderdale, FL: J.Ross Pub., p. 8, 2009.

[21] H. Beyer User Centered Agile methods, San Rafael, CA: Morgan & Claypool, p. 4, 2010.

[22] J.R. Hauser & D. Clausing. (1988) "The House of Quality," Harvard Business Review, pp. 63 -73, May-June 1988.

[23] Standish Group International, CHAOS Summary for 2010 retrieved 2013.08.17 from http://insyght.com.au/special/2010CHAOSSummary.pdf.

[24] C.G.Cobb. Making Sense of Agile Project Management: Balancing Control and Agility, Hoboken, NJ, USA: Wiley & Sons, Inc., 2011, p. 114.

[25] J.L. Cooke, Agile Productivity Unleashed, IT Governance Publishing, Cambridge, UK, p. 109, 2010.

[26] J. Tata and S. Prasad, "Team Self-management, Organisational Structure, and Judgments of Team Effectiveness," Journal of Managerial Issues, Vol. 16, No. 2, pp. 248-265, 2004.

[27] S. Koch, "Agile Principles and Open Source Software Development: A Theoretical and Empirical Discussion," Proc. of 5th International Conference on Extreme Programming & Agile Engineering, pp. 85- 93, 2004.

[28] D. Rawsthorne, Patterns that make Scrum work: Understanding and Scaling Scrum, LeanPub.com, p. 16, 2013.

[29] C. Khalil, V. Fernandez and T. Houy "Can Agile Collaboration Practices Enhance knowledge Creation between Cross-Functional Teams?," Proc. of the First International Conference on Digital Enterprise Design & Management (DED&M), pp. 123-133, 2013.

[30] L. Williams & R. Kessler, Pair Programming Illuminated, Pearson, Boston MA. p. 177, 2003.

[31] H. Kniberg & M. Skarin, Kanban and Scrum: Making the Most of Both, C4 Media Inc. USA, p.9-10, 2010.

[32] A. Martin, R. Biddle & J. Noble, "The XP Customer Team," in Proc. of the Agile Conf., pp. 57-64, 2009.

[33] J. Shore: The Art of Agile Development: The XP Team, Sebastopol, CA, O'Reilly Media, 2007.

[34] N.B. Moe, T. Dingsøyr, T. Dybå, "Understanding Self-organizing Teams in Agile Software Development," Proc. 19th Australian Conference on Software Engineering, IEEE, pp. 76-84, 2008.

[35] R.A. Guzzo and M.W. Dickson, "Teams in organisations: Recent research on performance and effectiveness," Annual Review of Psychology, No. 47, pp. 307-338, 1996.

[36] L.A. Griffin, "A Customer Experience: Implementing XP," in XP Universe, Raleigh, NC, USA, pp. 195-200, 2001.

[37] H. Robinson and H. Sharp, "XP Culture: Why the Twelve Practices both are and are not the Most Significant Thing," in Proc. Agile Development Conference (ADC'03), pp. 12-21, 2003.

[38] J. Recht and M. Nielsen, "Discipline in Extreme Programming," Retrieved 2014.03.01 from http://braindump.dk/dat8-discipline.pdf . p. 3.

[39] P.D. Leedy and J.E. Ormrod, Practical Research Planning and Design, New Jersey: Prentice Hall, p. 179, 2005.

[40] P.M. Nardi, Doing Survey Research: A Guide to Quantitative Methods, Boston, MA: Pearson Education, p. 119, 2003.

[41] C.A. Mertler, Action Research: Improving Schools and Empowering Educators, California: Sage Publications, California, p. 95, 2006.

[42] D. DeVaus, Surveys in Social Research. New South Wales, Australia: Routledge, p. 5, 2002.

[43] K. Conboy, "Agile Methods: The Gap between Theory and Practice," in Proc. 5th International Conference on Extreme Programming & Agile Engineering, p. 316, 2004.

[44] T. Addison and S. Valabh, "Controlling Software Project Risks – an Empirical Study of Methods used by Experienced Project Managers," in Proc. SAICSIT, Port Elizabeth, South Africa, pp. 128-140, 2002.

Aspects of Modelling and Processing Complex Networks of Operations' Risk

Udo Inden
Cologne University of Applied Sciences (CUAS)
Cologne, Germany
udo.inden@fh-koeln.de

Despina T. Meridou, Maria-Eleftheria Ch. Papadopoulou, Angelos-Christos G. Anadiotis, Iakovos S. Venieris[b]
School of Electrical and Computer Engineering, National Technical University of Athens
Athens, Greece
{dmeridou, marelpap, aca}@icbnet.ece.ntua.gr
[b]venieris@cs.ntua.gr

Claus-Peter Rückemann
Leibniz Universität Hannover / Westfälische Wilhelms-Universität,
Münster, Germany
ruckema@uni-muenster.de

***Abstract*— "Landscape of risk" (RL) is a metaphor to describe agglomerations of interdependent risk. The idea is to integrate the full scale, variety, velocity, variability and the related determinants of a complex operations' system into one computable model. The atomic elements of this network are managed nodes being exposed to risk, thus becoming source or target of unplanned events, of positive or negative impacts and propagation effects. Management is understood as continuing effort of operations' intelligence to realise and evaluate risk and to effectively act on it. The challenges are vast increases of the resolution of object and time and the accelerating change, of particularly technological innovation. These are reasons that RLs become more and dynamic, that models need to identify and capture interdependency across local and global levels and life-cycles, that learning needs to be directly integrated into the managerial workflows. Therefore, the RL concept allows for the integration of the "Big V" of data (volume, velocity, variability etc.) as well as for human and machine intelligence respectively learning. We discuss various problems and alternative models as well as architectures for processing complex landscapes and provide a first formal semantic model about the managerial handling of risk of for the management of unplanned events.**

Keywords-Integrated risk management, resolution of object and time, semantic models and technology, high-end computing

I. INTRODUCTION

A first and shorter version of this paper has been issued for the INFOCOMP Conference 2013 in Lisbon [1].

Landscapes of risk (risk landscape, RL) describe agglomerations of interdependent risk in business operations. *Risk* is one of the most general concepts of managerial decision making and capable of integrating a large variety of aspects into a coherent model of managerial acting. Of specific interest are the risk of occurrence of an event (*event risk*), positive or negative *impact* conditioned by this and strategies to learn from managing risk and impact. Figure 1 depicts basic views:

1) In the most general form, an RL is a network of interdependent nodes, each being target and source of unplanned events, i.e., of risk and impact (Figure 1-1). *Unplanned events*, discussed in Section V, are main issues of managing risk. The interdependency of nodes refers to *impact*, thus to conditioned probability as well as to learning. Figure 1-1 also differentiates autonomous (active) nodes disposing of managerial capacity as well as passive sub-nodes that are managed but may be a relevant resource.

2) Figure 1-2 describes an RL as complex supply chain, that is a distributed product-production-system (PPS) with the common goal to deliver material products or services to other businesses and finally to consumers. This network is designed according to the specifications of the PPS, as well as to the costs or availability of required resources. Impacts propagate along related dependencies.

To reduce complexity, supply-chains are typically defined on the level of the main nodes, the factories, i.e., the technological, organisational or other details may be known but are not managed on that level. However, the reasons of many failures that affect the overall efficiency of the supply

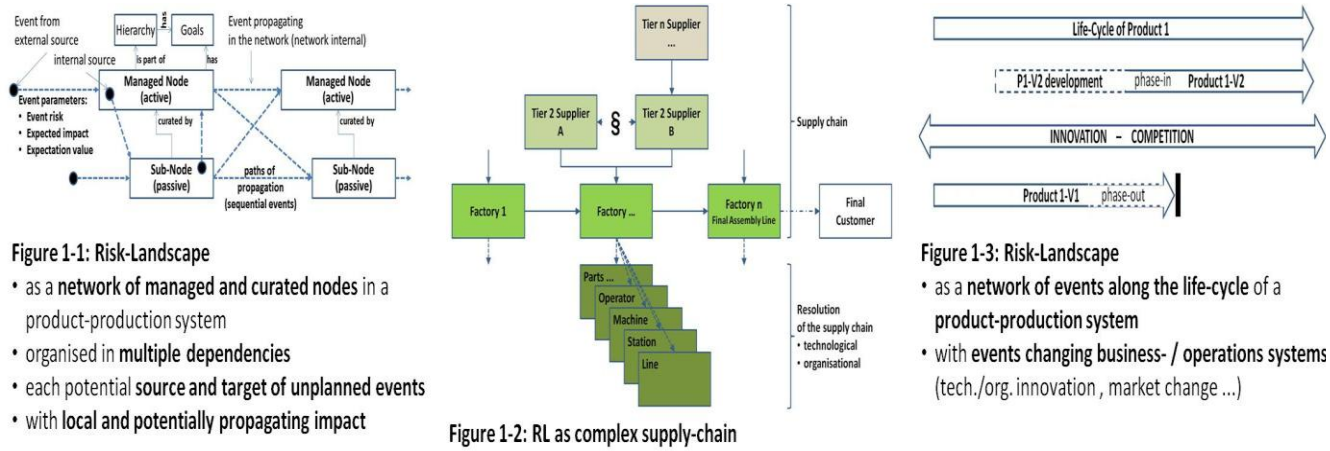


Figure 1. Three Basic Views at a Landscape of Risk.

chain actually lay in these details. Under conditions of close-to real time management, any detail matters, i.e., can become a target or source of unplanned events (see Section III-B).

In this respect, the former black box of the factory is resolved into production lines and possibly deeper into stations and machines in the production lines and, below, to teams or individual (named) operators. The Internet of Things also allows handling individualised resources like components or parts to be assembled. The final product of a car producer may not be the car, but the service it offers, i.e. the chain needs to include the details of a car-sharing system.

Not all of these "things" or operations are treated as active nodes (or only on demand). Nevertheless, the enormous heterogeneity of detail is the reason to increase the abstraction of models by risk-oriented concepts and by employing semantic models (Sections V, VI and VII).

3) Figure 1-3 presents the sequence of phases like the design and implementation / ramp-up, re-design needed to adopt a new technology, to respond to competition or, finally, to phase-out or terminate the product and the technology. In an RL-model, these phases translate into changing properties of nodes or deleting some of them in the network.

Learning is a major force in most of these phases: In the ramp-up phase the challenge is to overcome lacks of maturity of the design of the product or service and of the related production system (see examples in Sections I-C7 and II-B). In matured operations, it may become necessary to learn about options offered by new technologies or about the impact of competitors on the position in the market. RL-models will capture such changes or threads by adapting parameters that control risk or by adapting the structure of the network. So, the introduction of new additive production technologies will remove large segments of the suppliers' network and the related risk from the supply chain and from the RL-model.

In contrast to conventional models, RLs don't differentiate planning or simulation in operations or in life-cycle context on principle. It is always a network of interdependent risk and in the analysis of the vulnerability of an operations' system, even the same model that can be used.

The paper is organised as follows. Section II describes structural and dynamic aspects of RLs and Section III industrial research projects motivating the concept of RL. On that base, Section IV delivers a managerial framework and analyses forces that drive the problem of volume, velocity, variety, and variability in an RL.

Section V deals with knowledge models, continued by an overview of use of semantic technologies as well as Bayesian methods to gather and grow operations knowledge in Section VI, closing with a formal risk management ontology. Section XII drafts selected architectures to compute RLs, Section VIII gives an overview of future work.

II. On the Concept Of Landscapes of Risk

The idea is to integrate the full scale, variety, velocity, variability and the major related determinants into one computable model. The atomic elements of an RL are 'managed nodes' and 'unplanned events'. Nodes are the source or the target of unplanned events. Positive or negative impacts, as evaluated in the light of managerial goals, propagate in the network. Passive nodes are curated by active ones. RLs need continuous efforts to realise and evaluate risk as well as to act on it, while, on the other side, being challenged by an increasing speed of change and resolution of detail. This section describes structural and dynamic aspects of RLs.

A. On the Structures of Landscapes of Operations' Risk

Reference [2] states that "economics define investment as the act of incurring a cost in the expectation of future rewards". In business, risk is directly associated with success or failure of investment. Irrespective of the investor, the bottom line is that returns at least enable financial sustainability. This turns into rules, e.g., to maximize profit or minimize risk. If the environment changes, businesses need to adapt, i.e., profit is required to finance adaptation. Notably, change emanates from investment into innovation that implies risk, but proves its value as source of future income and, thus, triggers propagation in the markets, i.e., needs for courage to further innovate or adapt.

Figure 2 starts from investments in a business. To earn returns, functional domains with corporate (strategic, legal, financial affairs) or operational (engineering, production, purchasing) responsibility and related goals are allocated to managers (actors). Operations again are structured by processes and flows of data for planning, control and implementation actions (work). Typical passive nodes are potentially critical resources without inherent decision-making capacity.

Acting includes decisions (choices) and, thus, any actor has a managerial role in the reach of his responsibility, which is focused by related goals. This fits to a definition of Goshal and Bruch [3] of management as the "art of doing and getting done" in the reach of an area of responsibility, for given goals and related risk.

1) Managerial Responsibility: The concept of managed nodes implies actors who do not just take responsibility for the decisions they make, but also for their ability to make decisions in their specific position organisation (Figure 2), that is primarily for the access to relevant data and to information that provide context. Typically, there are downstream flows of decisions and information that provide context as well as upstream flows that enable to track and evaluate the progress of work and its impact. Horizontal exchange enables the coordination of work on the same level of the hierarchy.

2) Strategic decisions: The corporate level mainly handles risk related to corporate integrity, business model and strategy or financial sustainability. Capital bound in operations is the bridge between strategic and operational action. Within this framework and considering main parameters of the environment (e.g., issues in markets, life-cycle of products), operations' strategies specify the implementation of the business strategy into a consistent operations' strategy including objectives, accoutrements and collaborative workflows between operations' domains, or decisions on insourcing and outsourcing, choices of technology, etc.

3) Tactical decisions chose operations' policies (best practices) for given strategies or in sales with pricing or discount schemes. On this base, plans and schedules are elaborated to synchronize and optimize activities and flows of orders and materials, the provision of capacity and the

rostering of staff as well as take precautions for known contingencies. Finally, these plans have to be implemented, i.e., executed, monitored and, in case of unplanned events, to be maintained or recovered ("firefighting").

4) Work-level decisions: In wording of Goshal and Bruch, firefighting (or "educated improvisation") can be defined as the capability and capacity of accomplishing a goal, in spite of unplanned events, taking, though, the dependencies in the RL into account. The rationale is to maintain or recover active plans with minimal interventions. The speed of propagation of impact and the time left to effectively act are constraints to decision making. Thus, the lowest level of management runs in an 'exception mode'. Virtually, all responsibility concentrates in this node of action and the horizontal coordination with peer-nodes along the lines of propagation.

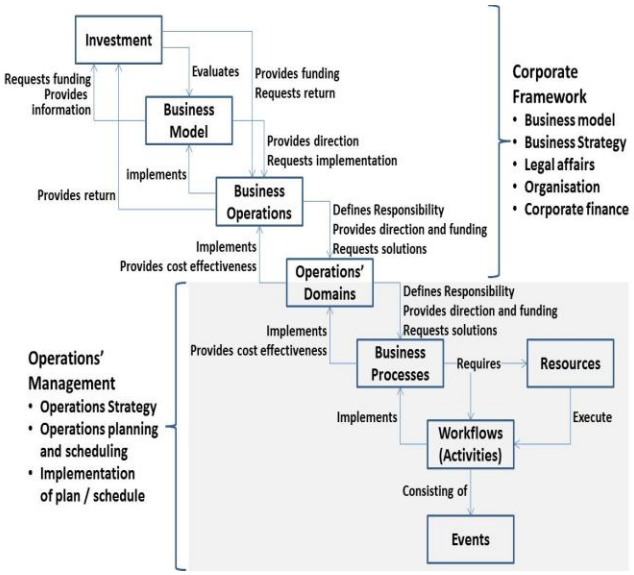


Figure 2. Locating Operations' Management in a Business Model.

5) Supra-network: Organisations depend on other organisations, e.g., across supply chains or service systems. Networks are not limited by organisational borders. In a wider scope, external nodes have to be considered. In a landscape of risk, such "external nodes" can represent a complete organisation, nodes in this organisation, etc. Structures and dynamics of this supra-network comply with Figure 1.

6) Dimensions of dependencies: Most actions have different contexts with different and potentially conflicting goals and dependencies. These contexts can be structured as a set of dimensions of managerial acting. For example, differences in the place may imply different legal frameworks. Most relevant dimensions are *organisation* (e.g., ownership, responsibility, hierarchy), *space* (the place of operations), '*time*' (aspects of synchronisation an performance) and '*technology*' (ways of performing activities). Further ones may refer to ratings in terms of solvency or product quality. The interdependency of dimensions is based on interactions between degrees of freedom in the domains. For instance, the introduction of containers offered additional degrees of freedom in global distribution, with further impact in other dimensions.

7) Structure determines function is a fundamental paradigm in many sciences. Propagation of impact follows dependencies and degrees of freedom in the different dimensions. For example, "not invented here syndrome" is about a problem that may propagate, e.g., along technical dependencies, but cannot be handled because of lacking responsibility, e.g., an organisational failure. Considering a dimension of different goals, the same event may simultaneously have negative and positive impact. An example may be a traveller arriving earlier at his destination by taking advantage from a delayed train.

B. On the Dynamics of Risk Landscapes

Events and their propagation transform the picture into a movie. In fact, propagation is the motivation behind risk landscapes. The identification and control of paths of propagation are major issues of the design of operations' systems. As an example, failing to avoid non-invented-here behaviour can convert paths into highways of propagation.

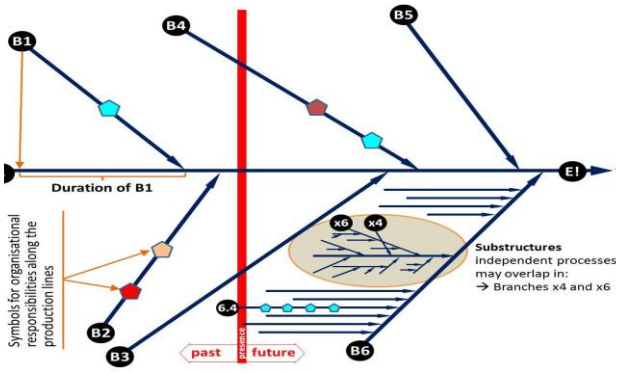


Figure 3. "Fishbone" Diagram for Cause-Effect Analysis (fictitious).

1) Paths of propagation: Figure 3 shows a version of a '*Fishbone Diagram*' [4] of a fictitious factory (for the construction of technical dependencies, the reader should refer to [81]). This diagram is an intuitive way of visualising basic cause-effect relations. A timeline is added, in order to show technological dependencies of a fictitious factory and, depicting the desired synchronization along processes in the work-breakdown. Parallel jobs are organised in different bones and sequential ones along the hierarchy of bones, while products move from left to right. The plant hosts six production lines with details shown for line B6. Each bone implies allocations of resources (materials, tools, and operators). Pentagons indicate different responsibilities in the process. Strategic change will have impact on these allocations, requiring taking measures to free or increase working capital.

A failure to assemble cable brackets in the fuselage of an aircraft can become the reason of a serious interruption of production, making the assembly of kilometres of cables in the next station impossible. Additionally, scheduled operations on the succeeding fuselages are temporarily stopped. Such defects also depend on structural decisions; in the case of the B787 'Dreamliner', Boeing finally decided to buy the suppliers that proved to be unable to solve the problems. The financial losses were tremendous [82].

2) Unpredictability and Non-linearity: A real RL is complex and exhibits an hardly predictable non-linear behaviour [5] that emerges from the number, the variety and the interconnectedness of acting nodes. The number of possible interactions equals to the power set of nodes, i.e., when external dependencies are to be considered, also small firms can generate landscapes of a very high complexity.

Non-linear behaviour appears in spite of well considered operations' standards. A major focus of risk management is to act on exceptions that may become critical in terms of the goals. From this point of view, standards of *firefighting*, so-called policies or best practices, are relevant for managing risk landscapes.

Using the Pareto's Power Law, 20% of unplanned events typically produce other unplanned events with 80% probability, while 4% of unplanned events cause 64% of trouble that may prove to be disruptive and translate into sizeable non-linear effects. Though the concept of the risk landscape is an abstraction, the actual complexity has to be captured. Lacking a comprehensive model, this can only be shown exemplary. Table 1 depicts parameters of events that structure these examples.

C. On Parameters of the Dynamics of Risk Landscapes

The dynamics of the RLs refer to the frequency and impact of change that may occur in different forms.

TABLE I. DESCRIPTIVE PARAMETERS OF UNPLANNED EVENTS

Type (aspect)	**Specifications**		
Type	technological	commercial	
Scale	size of business	reach of impact	
Decision hierarchy	strategic	tactical/implementing	
Competition	slow	dynamic	
Interruption	low frequency	high frequency	
Disruption	disruptive change	operations' disruption	
Knowability (simplified)	known	butterfly	black swan

1) The *type of change* here shall distinguish the technological and the commercial sides of operations. Both aspects, however, are interdependent: adapting to technological change may be inevitable, but the commercial impact finally decides about the sustainability of operations and business.

2) The dynamics depends on the *scale* of the related business (large firms/groups versus small/medium enterprises), simply because size implies a higher number of interdependent nodes and, in direction, a higher resolution of object and time (see Section III-B). However, the most important aspect of dynamics is the scale of *power*; the introduction of standard containers had high-power impact. With even more energy, the growing shareconomy, 3-dimensional printing (3DP), Internet technologies, business platforms and data driven businesses today change the rules in actually all industries worldwide (see Section III-C).

3) The hierarchy of decision making refers to the reach of events. For example, a new technology, like 3-dimensional printing (3DP), has impact on corporate decisions (see Figure 2). However, with the acceleration of the rate of change, the efficiency of hierarchies disappears. Finally, in firefighting scenarios, well managed firms enable low level actors with local knowledge to make decisions, even if they have strategic impact [93].

4) Competition boosts complexity by linking players into feedback circles: Challenges issued by competition need answers that may become an issue for competitors in the future. These cycles run in respective markets fast and resourcefully and imply a competition for innovation and related internal and external financial resources. Not at least it comes with significant risk. The loops can become even more complex. The "cooperate to compete" (coopetition) strategy exploits the competitive intelligence to make a cake together and, then, to compete for its pieces [6].

5) The Interruption of the progress of ongoing operation is the "lowest" level of negative impact of an event. It may occur if a critical resource is missing. For example, in 2010, the eruption of the Eyjafjallajökull volcano in Iceland temporarily stalled airline operations. Non-polluted air was the critical resource, analogously to the missing bracket in the scenario described before. However, an interruption can also leverage local noise to the level of strategic issues.

6) Disruptive change forces to discontinue a way to operate (the 3DP example) or to change or close business models. Interruption is a failure that can be solved, but may spiral into disruption (CargoLifter case). The cause may be a loss of a critical resources, external events, innovation or behavioural change. In the example of a shareconomy, customers deprive operations of valid business models.

7) Knowability refers to the chance to predict a development, that is, to recognise a potential event and its relevance. It is directly related to intelligence and acts as a prerequisite in the case of answering or driving creative destruction. Knowability implies abilities to estimate the expectation value of an event defined as impact multiplied by the event risk (see Sections III-A and IV-A). The question is what could be or can be made known early enough to reasonably act on a risk. 3D-printing is an example for issues that can be known. In this context, Bayesian inferencing can be applied in order to systematically improve such knowledge (see Section IV-B). Two further aspects that need to be taken into account, the Butterfly Effects and the Black Swans, are described below.

Butterfly Effects emerge from the non-linear behaviour of systems; small, hardly discernible causes lead to a large impact in hardly traceable ways. The analysis of large sets of data can support learning and a deeper understanding of the behaviour, i.e., the identification of positive feed-backs that fuel non-linear behaviour and propagation of impact. They are relevant in case of measuring the *criticality* of operations. In organisational context, a phrase in a contract can make a difference.

In contrast, *Black Swans* [7] come from places behind capabilities of imagination, at least for the vast majority of actors. Examples of strategically relevant Black Swans may be a sudden breakthrough in quantum computing or technology providing clean, safe *and* cheap energy. For most actors, the 2007/2010 financial crisis has been such an event.

Examples from direct operations are the problems in the ramp-ups of the Airbus A380 production in 2006-2008 and

of the B787 Dreamliner in 2003-2011. In the case of the A380, the highly customized harnessing became a problem because many of the cables were too short. It was beyond the capability of imagination and, thus, of awareness of all actors that different departments worked with different versions of design software.

The B787 case is marked by a long list of various disruptive events across major systems of the whole aircraft that, at least in this accumulation for management, were hardly imaginable. The disasters turned into additional costs of 6.1 Billion $ for Airbus and into an estimated 12 – 18 Billions $ loss for Boeing. For more details, the reader should refer to the "Catalogue of Catastrophe" [8] [9].

Neither a SWOT analysis nor, e.g., a simulation-based analysis will find a Black Swan. A possible solution would involve a systematic effort of *explorative learning*, of encouraging and facilitating creative lateral thinking or of taking advantage from diversity in the teams rather than fostering uniformity. For organisations and for actors, it is an act of balancing the discipline to act in conformity to standards or agreed proceedings on one side and on the other the intelligence of questioning standards and proceedings as potential habitats of Black Swans.

III. PREVIOUS WORK

The concept of risk landscapes, as it is discussed in this paper, goes back to a number of intra-industrial as well as collaborative research projects with industry. They delivered the empiric base of the concept. The major work includes air cargo logistics, selected airport ground operations, inflight catering systems, complex technological ventures, and, currently, small series production in aviation industry and work with a group of small enterprises on innovation strategies.

TABLE II. OVERVIEW OF STUDIES

Interactive Tracking	1995/96 Volkswagen, LH-Cargo
CL Knowledge Integrator	1998/2001 CargoLifter Project
RFID-based intelligent inflight catering	2007/10 Airbus (main partner)
Production Management	2012/15 Airbus, Iacobucci
For Fife SME – innovation strategies	2014/15 Group of SME

Adaptiveness to unplanned events, including the management of related risk and impact, has been the recurrent theme. The very first project became the primer. The idea to integrate virtually any aspect of acting under uncertainty into the concept of risk landscapes and realising the impact of accelerated disruptive innovation emerged from the projects described in this section (see Table II). The use of semantic technology and multi-agent systems was an early choice.

A. Interactive Tracking

In 1995, the Strategic Research Team of Lufthansa Cargo and Volkswagen Transport (VWT, the transportation unit of Volkswagen) agreed in a project that analysed methods to improve the response of the factory in Germany to urgent orders for spare parts by satellite factories. The work was done in collaboration with a team from the Technical University of Braunschweig [12] [13].

Satellite factories (in the study located in Mexico and in Johannesburg) assembled Volkswagen cars. Most of the parts were produced in a German factory and, by standard, sent by ship to their destinations, being too slow in case of emergency. To both destinations, the factory-to-factory air-transport time was about one week with five flights offered per week. Thus, in case of required response times to an emergency order of two to three weeks, up to ten flights could be used. The question was *how to exploit this flexibility to improve the flexibility of the customer*.

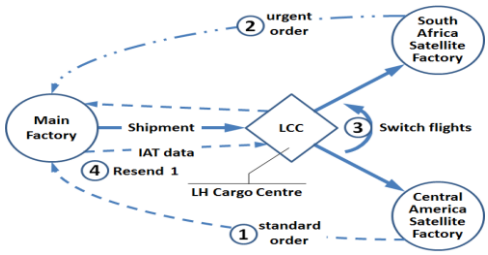


Figure 4. Interactive Tracking, Functional Scheme.

A typical case (Figure 4) is a request for parts ordered by the factory in Mexico. After packing in the main factory, the parts have entered the Lufthansa Cargo transport pipeline. From that moment on and until it arrived at its destination, the shipment became "invisible". Just after the shipment to Mexico "vanished", an order from Johannesburg arrived asking for a similar, but not identical, mix of parts. However, due to the fact that the production programs of the satellites were similar, the same applied to emergency demands. To avoid interruptions in the production line, the shipment should arrive the same week.

Since RFID was not yet available at that time, the idea was to mark relevant shipments using a barcode, in order to make shipments fully traceable for Volkswagen. Allowing for simplification, the process involved labelling shipments in the first Lufthansa station, and, then, scanning and storing there, as well as in any subsequent station, respective data, such as the airway bill, the place and time etc. The aforementioned data were maintained in a disk that accompanied the shipment until it was loaded into the actual flight. Provided that the staff was properly trained and disciplined, this process allowed to find and redirect shipments on order of VWT to the destination with the higher urgency; in Figure 4 to Johannesburg.

The management of operations' risk was not an explicit issue in the project. However, we were aware that logistic systems, able of handling a higher level of detail and a faster response to problems, can significantly increase operations' performance. It was a strategy to manage a "landscape of risk in the nutshell". Together with the Kenan Institute of Private Enterprise (University of North Carolina), the idea was picked up by an international researchers' network [12].

B. CargoLifter Knowledge Integrator (1995/2002)

The CargoLifter Project (CL, Germany 1995-2002) aimed at designing, producing and operating airships of a size of 260 m. length, 65 m. width and 82 m. height: a 'flying crane' for transporting goods of up to 160 metric tons up to 8

by 8 by 50 meters in size. The project failed because of its overcomplexity. It included the job of an airframer (design and production), of an airline (flight operations), of a ground infrastructure provider (each parking position of this airship would require about a square kilometre) and of a logistics company specialised on complex special transport projects [14].

To substantiate and justify investments into this project, a major challenge was to intelligently link the task to establish a valid business model and the task to support the acquisition of financial resources on one side with the technological progress of airship development on other side. In order for this to be accomplished, the strategic research team of CL specified the '*Knowledge Integrator*' (KI, realisation by Magenta, London and Prof. G. Rzevski, Open University, Milton Keynes).

A first purpose was to estimate the financial performance of airship operations' networks including airship, infrastructure, customer sites, orders, etc. as nodes. The specification of the market relied on different sources: (1) real data from members of a global group of industrial lead-user about their projects, (2) data from global market and benchmark studies, and (3) even potential competitors like shipping lines, and (4) by the airship engineering team providing estimates parameter values of the overall airship operations' performance as specified. All data were regularly updated.

Thus, the model *integrated* the knowledge available about performance parameters estimated on the base of the progress of technological and operations' design, various market studies and on knowledge inhered in questions of investors, lead-users, banks, authorities, press, etc. Based on that pre-knowledge a Bayesian process was started by specifying and simulating operations scenarios with the goal to deeper understand and improve knowledge a posteriori. In this way, technicians stepwise improved their estimates of the performance of the airship and its impact while share- and stakeholders reviewed their ideas about expected returns or the capital to be bound in development and operations.

Above all, the idea was to systematically grow the KI to the capabilities of a fully-fledged operations' management system dynamically acting on unplanned events. Although the term "risk landscape" was not used at that time, the model fully meets the definition. In the bankruptcy of the CL project, the developed software and the majority of the documentation unfortunately perished.

C. RFID-based Intelligent Catering Systems (2007-2010)

iC-RFID was a strategic research project, funded by the German Federal Ministry for Economic Affairs and supervised by the Program Management group of the German Aerospace Centre [13]. The purpose of this collaboration of five industrial partners and three research institutes was to design and demonstrate functionality and business cases of end-to-end integrated RFID-based inflight catering systems.

CESAR (Configuration and Evaluation of Service Systems in Air-Catering with RFID, [15]) was one of the subprojects, designed and implemented by the research team at Cologne University of Applied Sciences. It was a prototype of a multi-agent system integrating major RFID-enabled functionality of inflight catering, novel service models and further technological innovation. The model included the specification of catering services by airlines (food, beverages, sales items etc.), production and packing service content by air-caterers, selected airport ground operations, ground transport and exchange services (highloader trucks for unloading and loading aircraft galleys) and main aspects of the rotation of service equipment, such as trolleys. At the same time, the flights in the airline networks, standards of aircraft producers and of particularities of aircraft interiors, such as galley layouts and equipment, as well as legally enforced regulations were considered.

The MAS assisted the management to maintain service levels in case of unplanned events by analysing discretions to act of all active nodes in the scene, by proposing solutions, organising the implementation of decisions and tracking the effectiveness of action. However, *CESAR*, a prototype of a context-sensitive real-time operations system, was only able to respond to events that had already occurred and not to the risk of events. Methodology, such as tracking of criticality and evaluating event risks on the basis of behaviour, enabling the uptake of proactive action, was not implemented.

D. Adaptive Production Management (2012-2015)

ARUM is a collaborative project with 14 partners. The ARUM project concentrates on two use-cases from the aviation industry about the production of aircrafts and aircraft interiors. ARUM is co-funded by the European Union in the 7th Research Framework Program (GA 314056) [15]. One of the use cases focuses on the *ramp-up* of production, which is one of the most critical phases in the life-cycle of a complex product like aircrafts. Stories about the Airbus A380 or the Boeing B787 'Dreamliner' are known [8] [9]. These ramp-ups are marked by possibly fatal problems of technical maturity of components and processes as well as by poor learning curves as a consequence of the small series in aviation industry. ARUM provides MAS-based planning and scheduling systems that capture and process unplanned events in large scale and are prepared for risk-sensitive methods.

Airlines, as the final customers and operators of aircrafts, expect innovation that saves costs or improves services. The trend is to pack them into refurbishments of existing programs instead of ramping-up all new options with the start of a new aircraft program. Thus, ramp-up scenarios will also appear during the lifecycle. In this context, colleagues of the National Technical University of Athens (co-authors of this paper), as well as from Manchester University, Certicon, Prague and CUAS among others coordinate the development of semantic models (ontologies) [1] [16] [89].

In this work, it became clear to us that accelerating change is *the* pervasive force driving issues like repetitive ramp-ups, challenges to managerial workflows, further fragmentation of learning curves, and not at least inconsistencies and losses of effectiveness of semantic models. Moreover, it is the source of disruptive change and increasing unpredictability. In collaboration with Almende, Rotterdam, strategies that enable managerial workflows to keep pace with the quickening of technological change and that effectively support learning strategies will also be implemented.

E. What Matters: Lessons Learned

1) Agility matters in terms of adaptiveness to any change. The IAT project was motivated by ideas of an industrial Agile Management program [17] [18] that focused on impacts of fast action on unplanned events and of a clear dedication to the customer in terms of "We learn and solve your problem!". Regarding the reduction of uncertainty about the progress of a complex project, the Knowledge Integrator was inspired by this idea. It generated verifiable information that reduced risk and, thus, increased value of invested time, knowledge and capital. The iCRFID project targeted the exploitation of RFID and, in direction, of the Internet of Things, for planning and ad-hoc exception management. ARUM is about increasing adaptiveness and reducing uncertainty in the ramp-up of complex product-production systems.

2) Detail and local particularities matter. IAT tackled a very small fraction of VW shipments. However, the expected loss (impact) of any of these problems exceeded by far the efforts. *CESAR* again calculated potential benefits of corrective action. Both solutions intelligently handled details captured on the lowest level of sensing and acting in local operations. A similar approach is taken in the ARUM project.

3) Precision of mapping matters. It was unlikely that a shipment, deviated from Mexico, contained *exactly* the same mix of parts as ordered by the Johannesburg factory. However, it provided an intermediate solution. On the contrary, iCRFID and ARUM do not allow for variety. For instance, to ensure "delivery as promised" under all circumstances, item X, passenger Y or aircraft Z became active nodes represented by software agents that track and manage the way from a storage to the seat of the passenger. In case of failure, the affected agent issued a request for corrective action, e.g., by taking a spare item from the next source capable of solving the problem.

4) Multi-agent technology matters. Depending on the size of the scene, thousands of details may matter. At any time, each can become the critical resource and the change of any parameter may devalue an existing solution. Potentially fatal chains of events (propagation effects) or positive feedback circles that are caused by the behaviour of particular nodes or sub-nodes are additional reasons for stressing the need for capabilities to model and process objects and events on the lowest level of operations and detail. Therefore, in spite of limitations, particularly with regard to their potentially high load of communications, only MASs have proved to fully satisfy requirements to control ongoing operations under the condition that a particular detail here and now has impact on another particular detail and that solutions depend on the discretions to act that are available on that level.

5) Semantics matters, as operations systems can exhibit a high and dynamic heterogeneity of objects, processes, frameworks and terminology. Semantic modelling seems to be the only solution providing the flexibility and adaptability that effectively supports the R.E.A.L. processes.

6) The quality of managerial workflows matters. Effectiveness and efficiency of operations ultimately rely on efforts to systematically adapt and improve managerial workflows. In changing environments, this is a continuous task. This is the core of managerial excellence as defined in the Beste Fabrik program, active since 1995 and based on far more than 1000 industrial case studies in six European countries and run by seven major business schools [19]. In ARUM, we identified concrete needs for improvement as prerequisites of any effective use of intelligent tools. In the CargoLifter project, deficits of management workflows became the major reason of bankruptcy [20].

IV. A Framework of Managing Risk Landscapes

A. R.E.A.L. – *Realise, Evaluate, Act, Learn*

Realise – evaluate – act – learn is a generic logical structure of proceeding that describes the behaviour of managed (active) nodes in RLs. These tasks are not trivial, either on the local level of individually responsible managers or on any level of integration, and particularly not in case of response to unplanned events. Even ideas may be lacking about what actually has happened and how to proceed further. Without question, managerial effectiveness relies on human factors like discipline and readiness to act, high attention with a sense for details *and* the big picture as well as on high communication skills. The poorer the management is, the less likely it is to keep pace with the propagation of events. As a result, the management fails to mitigate drawbacks or to take advantage from upside potentials.

1) Realise: Nobody can act on unperceived events. Perception may fail because of lacking training or lacking attention owed to human shortcomings. An event may not always be recognised and, thus, not communicated. Sensors may fail or be missing, signals may be filtered out or an event may be not properly read or vague in its meaning and need more knowledge to be understood [7].

2) Evaluate: The decision whether or not to act on *identified* unplanned events depends on thresholds of its relevance. In economic contexts, the relevance is expressed by their expectation value, which is the product of event risk and impact, with the event risk p of an unplanned event that has occurred being equal to *1*). In regulated environments like aviation or health industry [21] [22], particular classes of events may be relevant by default in order to avoid quality hazards. If events and their contexts are clear, evaluations can be supported by ARUM technology or Big Data applications, and, in standard scenarios, possibly be automated.

3) Act: Acting on unplanned events (re-)establishes planned states by implementing a suitable policy. If there is time and planning capacity the plan can be updated. In some cases, rules or a proven best practice may be applicable. Elsewise it needs "educated improvisation", e.g., of an experienced dispatcher and the hope that it works.

4) Learn: Unplanned events are the reason and the resource for learning. Deep knowledge about a system derives from enduring observation of its behaviour in a large variety of operations' scenes and the review of many failures along R.E.A.L. processes. In case of disruptive change or innovation, contexts of learning may be lost, "old" technological and managerial knowledge may be devalued and new learning curves might start. It is very likely that experimental learning will have to support or even to replace practical experience. Nevertheless, while experimental learning analy-

ses the behaviour of complex but widely known systems like a factory, a supply chain or a service system, explorative learning tries to avoid dependency on current knowledge [23]. It is far more permissive and allows for testing ideas that may be very strange in the eyes of domain experts. Ambiguity and complexity here are resources. De-learning is becoming a topic. The focus shifts to the management of transitions and the identification of re-useable knowledge.

Learning as a continuous effort is the backbone:

- *Operations' Intelligence* is the capability to effectively disambiguate complex scenes in all phases of R.E.A.L.
- *Real-time operations' control*: Many events need immediate action to answer to downsides or upsides.
- *Tracking of effectiveness*: The effectiveness of implemented policies has to be measured and analysed. And new events require further action.
- *Awareness of assumptions* is a core aspect of learning, like in Bayesian experiments that explicitly capture prior and posterior knowledge (see Section V-B).
- *Encouraged and augmented learning*: As failures and "strange ideas" become sources of learning and innovation, a culture needs to be developed, in order to elicit rules and to provide resources particularly for learning from failure and exploration. Organisational structures form the base of effective augmented learning, including the effectiveness of computer-based support, such as simulation programs. Carefully explored and deployed *data-driven business and operations intelligence* are about to become a further opportunity of learning [94] [95] [96].

B. On Interactions of Forces Driving the Big V and on Related Control Problems

From containerisation to servitisation and 3DP, the origins of the Big V are interplays of technological, economic and social developments that also drive the phenomenon of acceleration. Modelling interdependent risk needs to conceive and decode the driving forces and their impacts. From this point of view, velocity, variety or variability does not form the problem; their increase is. More importantly, the situation deteriorates with *acceleration* that hardly leaves a chance to accustom to a plateau or a rate of change. These accelerators are inherent to relations between nodes. Namely positive feedbacks, are relevant and, in consequence, resources that enable management to strategically and operationally control accelerating processes. Multiple facets of the accelerators need to be considered in a model.

1) The globalisation of the reach of almost any activity, the increasing informational connectivity of everything via the internet, the abstraction of businesses and operations as well as the competition by digitalisation are the driving forces. They are inseparably intermingled and locked into multiple amplifying feedbacks (Figure 5). Historically, the development is a stepwise facilitation of the exchange of everything by converging technologies; containers for the physical part, the internet for information and data as well as the virtualisation of services and, finally, the digitalisation that adds computability.

In terms of risk landscapes each of the phases boosts:

- the volume, variety and variability of actors (managed nodes in Figure 1),
- the resolution of these objects (sub-nodes, i.e., further detail, things or services), producing more volume, variety and variability that are relevant in terms of goals and risk of acting,
- the resolution of time, i.e., a higher frequency of unplanned events, particularly of two kinds: (a) creative destruction and (b) operational risk.

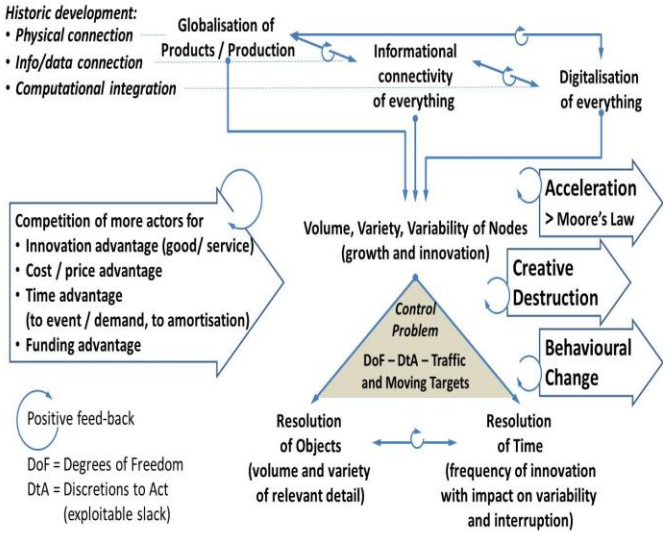


Figure 5. Driving Forces of the Big "V" and of Accelerating Change.

2) For given feeding grounds, *competition* is driven by the number of competitors, the information available to compete and the capabilities of mining, selecting and processing relevant data. Thus, competition shapes these developments by positive feedbacks. Basically, there are four ways to compete: (1) for the better product, (2) for lower costs and lower prices, (3) for speed of acting, and (4) for the access to capital that is required to pre-finance innovation or to cover risk. The choice of these strategies highly depends on the current conditions and the involved actors are seldom able of playing all combinations.

3) The resolution of detail may increase because technology consists of more parts, but also decrease with new technologies like 3DP. The problem, however, is not the volume but the potential criticality of detail and their managerial impacts [83]. In industry, more detail implies more types of stock keeping units (SKU), more supply chain complexity, and larger stocks, i.e., more working capital.

Thus, financial departments want less, while operations departments like to be on the safe side. Competing for access to capital, the winners are clear. In the IAT, project a VW manager answered a question about benefits: "*We expect a reduction of shipments and volumes of materials in transport. In the long run, we may also be able to reduce inventories*". That is to say, reduced inventory consumes improvements of control and, in consequence, more details become more and more critical.

4) Ultimately, *competition accelerates* because time always matters; the time to amortisation of an investment is crucial to capital cost. Innovation needs timing to the market and, at the same time, it acts as a force of creative destruction and disruption. Response time to demand is a major factor of service quality. Shorter lead and cycle times require to invest into process innovation and form programs eliminating cost drivers, working capital or reducing the time to amortisation, i.e., the time capital is committed to a particular business and therefore under risk [24] (see Figure 9).

Time competition is a primary and acceleration a secondary effect of the forces and the underlying economics of capital turnover. Moore's Law is not the issue; its impact is; hardware capacity doubles every two years, but the richness of exploiting this capacity grows faster. The same applies to behavioural change, e.g., the demand for service.

In consumer markets the demand for technology has been educated by hardware providers, like Apple, grounding their business model on short product life-cycles. Therefore, not only competition and market but also the life of people and turnover time of fashions or trends accelerate and, in the eyes of customers, change the focus of utility.

5) The Ashby problem: Asby's Law of Requisite Variety [25] requires controllers to dispose of at least as many degrees of freedom (DoF) as the system that must be controlled can exploit to produce uncontrolled (unplanned) events. It is about the *variety of behaviour* and includes DoF available from different constraints in *different dimensions of acting*.

In this respect, a solution may not become effective in production because of organisational failure, like a non-effective allocation or handling of responsibility. Control relies on effective constraints to behaviour, as well as on effective policies to respond to the upsides or downsides of unconstrained behaviour.

Ashby's subjects of control were technical systems: "*if variation is required* [to control behaviour], *there must be a source of noise* [yet unknown DoF] *or information* [about uncontrolled DoF]" [26]. A strategic scenario could be a search for drivers of unexpected fluctuations of sales. In operations, these drives could involve variations in the quality of supply. Nonetheless, real customer relationships, competitive games, operations' systems and markets are not complicated but complex. They exhibit emergent behaviour and are populated by positive feedbacks that mutate butterflies into gorillas, ideas into creative destruction or responsibility into a "not-invented-here" syndrome. Nothing of this can be reduced to the behaviour of single nodes and strongly dependent on contexts and history.

6) In the "slow motion" environments of Ashby, there was not enough energy in the system differentiating complexity from complicatedness. Nevertheless, Ashby's Law still holds. In order to lock into positive feedbacks and to grow exponentially, there must be nodes in the RL that dispose of appropriate DoF. For their detection to be feasible in time, business intelligence needs to get onto the track of indicators and of potentials to change. It is an issue of operations' intelligence to get onto the track of early butterflies, e.g., by analysing noise.

7) Discretions to Act: Almost any schedule allows to accommodate another appointment. If need be, contact persons of booked appointments may be asked to shift or to wait a little.

8) Multi-agent systems in the CargoLifter or the iCRFID project worked in a similar manner; if a delayed flight blocked two catering trucks in peak time, no truck had a slot available to take over. However, it is often possible to shovel capacity free by managing small shifts across the fleet, and, if necessary, involve further stakeholders. The use of Discretions to Act (DtA) implies using slack in a system as a resource. This is based on two capabilities: The first involves understanding slack as a resource of flexibility. The second one is to identify and effectively exploit the DtA.

However, they cannot be planned, but they can be constrained or used up by scraping the bottom of operations' resources. Optimal slack can only be tuned based on experience and simulation analysing patterns of noise and needs for flexibility. DtA are also hard to track, since they are a volatile resource that disappears (the truck is stuck in a traffic jam) and re-appears because another problem, like a cancelled flight, shovels time free.

Centralised control will barely handle interactions of volatile DtA and dozens further context parameters. Solutions emerge from trading DtA, but they are local and bound to the slices of time available to the individual actors. It needs *peer-to-peer systems* (P2P), i.e., nodes that are aware of their current states and, on that base, coordinate their local, individual decisions.

Collectively they are aware of the integrity of the overall process [27]. Further the nodes can realise and coordinate their exploitation of DoF and their exposure to risk across the network, e.g., in the same way that a car-to-car communication system uses DoF of individual cars for accident prevention and traffic control [28]. So a minor reduction of speed may avoid an hour wasted in a traffic jam.

Vast volumes and velocity of communication is the price to be paid for the advantage of widely autonomous P2P systems to control operations and to deliver indication for strategic decisions. For reasons of comparison, the *CESAR* prototype included about $10^2 – 10^4$ nodes, whereas a realistic full scale model would need about 10^6 nodes.

For a network of factories, the ARUM model may reach the same dimension. The current European air traffic involves about 25 thousand flights per day and it is expected to reach about 43 thousands flights per day by 2030.

C. The Challenge of Accelerated Creative-Destruction

"*The paradigm shift rate (i.e., the overall rate of technical progress) is currently doubling (approximately) every decade; that is, paradigm shift times are halving every decade (and the rate of acceleration is itself growing exponentially). So, the technological progress in the twenty-first century will be equivalent to what would require (in the linear view) on the order of 200 centuries. In contrast, the twentieth century saw about 25 years of progress, since we have been speeding up to current rates. So, the twenty-first century will see almost a thousand times greater technological change than its predecessor*" [29].

J.A. Schumpeter defined innovation as a force "*that incessantly revolutionizes the economic structure from within, incessantly destroying the old one, incessantly creating a new one*" [30]. Investors see here a portfolio of options to commit capital to novel technologies and business models. In IT-driven markets, a system or a model may be characterised as "old" within a year. But what is "old" in an accelerated competition for innovation that vice versa is an accelerating force? What is the risk that options to innovate or to establish a business vanish the next day in global, internet driven competition with brain-to-market times being cut by 3DP? In the digital business of speed, trading the shelf life of information counts in milliseconds and technology to improve response times is implemented as it appears.

In terms of RL, it is crucial that any of the developments discussed in the following paragraphs had or has unescapable disruptive strategic and operations impact. Nonetheless, all of them have been knowables, and, thus, left time to adapt. With Big Data and 3DP, this fact also changes the border between strategic and operations' aspects of RL erodes with further acceleration.

The list of firms that failed to adapt to change is long. As an example, on July 30th 2014 the CEO of Osram, the second largest producer or electric lights in the world, said: "*The whole industry is taken aback about the fast decline of demand for traditional products*" [31]. The whole industry? The source of failure is shortcomings of industrial and operations' intelligence. In the words of Clayton Christensen, "*Outmoded thinking and the tyranny of key performance indicators impede innovation that creates new markets and new jobs*" [32]. A brief overview:

1) In less than 20 years *the Internet* became the global exchange platform for information, data and data-based services allowing to directly or indirectly connect and control virtually any 'thing' by sensors and actors. Aware of planned or actual states, any object can also be directly (via embedded IT) or indirectly (via software agents) become an active node in an RL (Figure 1). Equally, any digitizable service, either complex infrastructure or application, is available as a scalable cloud service. Based on semantic annotation or ontologies, the Semantic Web [33] gives bytes and data a meaning and facilitates internet-based knowledge processing and intelligence. Web technology became the mainstream of semantic modelling, of data filtering and integration as well as of data-based services. The Internet became the largest imaginable agglomeration of data and the delivery room of the "Big V".

2) This data cloud is a bonanza, available for mining and exploitation in science, business, and the public sector or state agencies. As the Big V, Big Data (BD) are children of the Internet. BD capabilities and capacities are unprecedented accelerators *by cutting the time from data to business* by means of scanning, organising and analysing massive volumes and flows of data. BD are breeding grounds of new scientific practices [97], of the scientification of businesses and of new occupational profiles, of new business models – and the doom of others.

3) The actual value of goods is equal to the services they deliver: a car that fails to start is no asset but a problem. The result is a growing shareconomy. Thus, *servitisation* is a strategy to enrich goods by services (e.g., cars by assistant systems) or, above, to sell not the goods but the service they enable, a view that implies that customers learned to evaluate products in a different way: A taxi is a *car as a service*, a capacity shared by taxi passengers, virtual and scalable in the sense of virtual computer capacity.

The number of people sharing instead of buying cars doubled since 2004. The ambitions may be too low, but by 2020 Mercedes plans one Billion Euros of revenue by car sharing (less than 1% of sales in 2012) and by 2030 BMW wants to make more money with data than with cars. Recent studies indicate that in agglomeration areas a shared car replaces up to 32 bought ones. The change of social values changes a global industry.

4) 3-D Printing (additive manufacturing) revolutionizes the production of most material goods including transplantable tissues and organs [34], food [35], fabrics, clothes, toys, dinnerware, buildings [36], and parts for aircrafts [36] [37] [38] [39].

"*The most exciting thing about 3DP is that complexity is free. The printer doesn't care whether it makes the most rudimentary or most complex shape*" [40]. Moreover, lot size equal to one is almost for free, large parts of tooling and a large variety of supply become needless, working capital is remarkably reduced and economies of scale are reduced to learning how to improve and operate printers [41].

Printing as a service will become a mainstream of production, provided close to place and time of demand. In fact, Amazon just announced the launch of a 3DP-Store for customizable goods [42]. More important, *3DP digitalises the way from brain to market*: the essentials are the file coding the design, materials that match functional requirements and a printing device capable of processing the materials in the right time and with the right quality.

Anybody able to create a new design or functionality the proceeding will be to experiment, learn, optimize, and sell. Additionally, times from science to business become shorter: printable materials with new properties or printers with new capability will immediately change options of design and production.

V. SEMANTIC MODELLING OF RISK LANDSCAPES

The scale of problems of "integrating everything" into a processible model requires an ambitious semantic effort, the more as the concept of risk landscapes does not reduce the heterogeneity or variability of operations and the need for managerial knowledge. It actually reduces the semantics to describe and analyse risk and its propagation to a few concepts: event risk, impact and the expectation value of events that controls the relevance of acting.

Risk-effectiveness relies on the performance to coordinate decisions across the network, and not on the effectiveness of nodes to meet local goals. Event risk and relevance are landmarks that help to navigate and to coordinate acting in the landscape. This section describes basic concepts of semantic technology, the potential parallels to observations and discussions in geosciences, and questions deriving from the unprecedented increase of the Big V and of the accelerating speed of change.

A. On the Subject of Semantic Models

Does *this* sentence have a meaning? Not on its own resources but if this paragraph provides it with a context. What is the meaning of water dropping on your head? Outdoors, it may be '*it rains*', whereas, indoors, it may imply that '*the pipe leaks*'. In the same respect, what is a lightning? Members of a primitive tribe may take it as an omen. In an invitation, the term '*ball'* may get its meaning by the request to wear a ball gown or a soccer jersey. Obviously, meaning lies neither in the terms nor in observations (perceived signals). It is in the context, that is, the network of associations on our disposal to accommodate the sentence, the drop or the lightning, in a non-ambiguous way that "makes sense" rather than irritates. In this respect, different observers of a lightning may use the same term but may not share the same interpretation, because they do not share the same knowledge model.

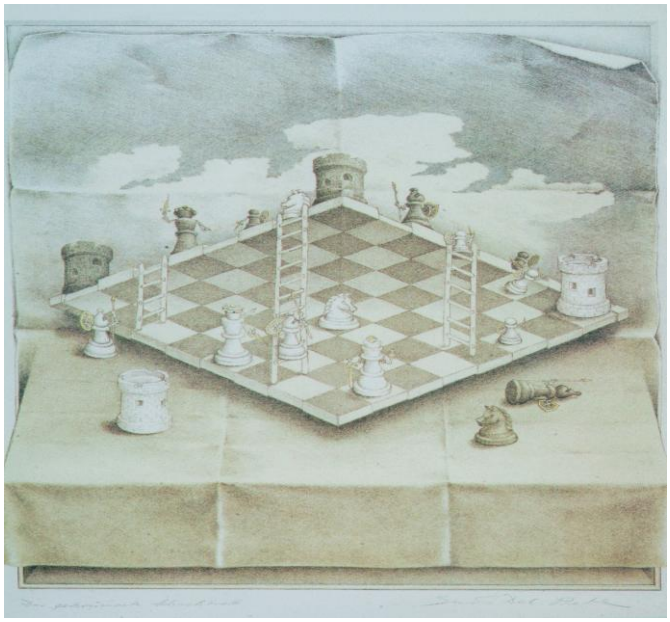


Figure 6. Sandro Del Prete: The curved chessboard [43].

A short insertion clarifying the use of terminology in this paper may be useful:

- *Perceived signs or signals* of any type indicate the arrival of an event and need interpretation.
- *Semantics* is the meaning of signs that emerges from associations to other meanings (context).
- *Concepts* are disambiguated meanings, i.e., have clear associations to other concepts. Conceptualisation is the process to disambiguate a network of associations.
- *Terms* are names for concepts. A definition is the disambiguation on the level of names.
- *Knowledge* is the network of concepts that enables an actor to act in a real or in an imagined environment. Sharing of knowledge requires a sharing of meanings and of terminology.
- *Intelligence* (also business or operations' intelligence) is the performance of adapting knowledge and new events to each other and to maintain or improve the consistency and to reduce ambiguity of knowledge.
- *Ontology* is a conceptualisation of being, pragmatically a model of knowledge about a domain that can be shared, based on an agreed terminology with an adequate precision and consistency.

Figure 6 illustrates, however, that the proceeding of perception, disambiguation, and acting may fail. Signs may have different meanings, qualitative data may tenaciously resist structuring and workable data may not be available.

This is at the heart of the management of RL that is not made from figures, but from the meaning of figures, events, or stories (narratives) that provide context [93]: why bought hedgefonds bonds of a bankrupt Argentina? How meaningful is the definition of a car as "immovable property"? Which knowledge is shared by the Osram CEO with '*the whole industry'*? How can we identify, and what is the meaning of a Butterfly or a Black Swan? All this is about contexts.

In terms of semantics: hedgefonds have a different ontology and value risk differently because they have a different business model, young urban citizens and traditional customers see different values in the same issues. The reasoning of Osram management was mislead by knowledge that did not match reality.

There is no value-free knowledge because any step in the R.E.A.L. process implies decisions and each decision relies on *currently available pre*-knowledge, that is preliminary knowledge built from assumptions and values motivating initial ideas that prime the proceeding and questions, e.g., to be answered by a Big Data application.

Butterfly effects or Black Swans frequently are not realised because they are beyond perceptions and beyond imagination or because they are in conflict to our fundamental assumptions and to values that organise our world. Therefore, the educated gut feeling of an experienced operator frequently performs better than any system.

In comparison with the CargoLifter KI program, the idea is to systematically mature the model by unifying the formulation of questions, growing the awareness of underlying assumptions, clarifying the terminology, understanding interdependencies of the domains involved and agreeing in a common ontology that enables effective collaboration. This is not far from the challenge to model and capture information in an RL. In contrast, environments marked by accelerated change and an increasing resolution of object and time imply repeating efforts of semantic re-engineering with phases of high or of greatly reduced matureness.

B. Potential Reference Strategies from Geosciences

In the first version of this paper [1], we agreed to use a model from geosciences as a reference. Lacking experience with modelling and processing complex risk landscapes it looks plausible to compare them with weather models of a continental scale. Interestingly, looking for examples for modelling and processing of very large volumes of heterogeneous data, we found relevant work in geosciences.

Managers must decide with the knowledge they have, which is an aspect that justifies a short review of Bayesian models below. Not making a decision is the worst decision not at least because it divests of learning and in practice not

"the" optima, but because best possible solutions are achieved. Kristine Asch describes this problem on the example of creating "*a harmonised geological dataset for the whole of Europe and adjacent areas*". To get an idea of the multiplicity of constraints of managing RLs, it would be useful to refer to [44].

Though they differ in terms of resolution or dynamics (probably except for the case of high-energy atmospheric processes), there is indication that geosciences and risk landscapes face similar requirements regarding semantic methodology. Both include large sets of numeric data and extensive non-numeric content including vague concepts or narratives to be disambiguated and formalised, because indication and evaluation of change often does not lie in the figures but in their interpretation.

The distribution of work of geo-scientists and managers in risk landscapes need interdisciplinary collaboration and depend on individual perception. Both need modelling to maintain the connectivity to lowest levels of possible data sources as well as to allow for the largest possible variety and the lowest practicable degree of reduction. This process needs terminologies that not unify but align variety.

In slow-motion environments, standards make work easier. But, this can hide indicators of change. Under conditions of accelerated change, the tolerance to some ambiguity may be useful. Nonetheless, interoperability is paramount in both areas. Upper ontologies serve as cross-mapping hubs for satellite ontologies that hold more specific local concepts. Reducing the ontological problem to a terminological one, a systematic restriction of interpretation is a way to reduce confusion [45].

Based on an example of the ARUM project, the specification of an element in a risk landscape could be 'bracket-is_a-resource' completed by 'resource-has-dimensions' and 'owner-is_a-dimension', 'bracket-has-owner', etc. Each step narrows the space of possible interpretations. The proceeding is handy and compatible with modelling RLs. It also allows for playing with constraints, e.g., for explorative learning or testing of alternative modelling strategies.

C. Allocating Semantic Modelling in the Organisation

For the semantic engineering of large risk landscapes, this suggests to establish networks of actors (nodes), gathering and modelling local data according to principles discussed above, including a sense for deviances from expected behaviour (contradictions to a hypothesis), a training issue. In practice, the task should be allocated to knowledge management departments.

Modern knowledge management departments are service centres offering products customized to corporate or operations' strategies and tasks. They capture and disseminate tacit and explicit knowledge, as well as encourage and facilitate the direct exchange in the organisation, e.g., via social media, linked data infrastructures, or personnel rotation programs and allocations to projects requiring collaboration of departments. They are engaged in formalising knowledge, in related internet projects, in content management and quality management programs for KM-services, content or effectiveness of knowledge. KM is involved into the support of Big Data applications, e.g., by developing semantic filters for heterogeneous mass data. Thus, KM is the most important candidate to implement n semantic infrastructure for operations' intelligence and managing RLs. Thus, the goal is that well educated knowledge managers assist operations' line managers and feed the model.

D. Open Questions – some Examples

In the following, we aim at exemplifying a few topics that may need further discussion. One of these issues is that risk landscapes, to a large degree, are concerned with human behaviour of perceiving, evaluating and modelling human behaviour. Problems of biases or path dependencies ("history matters") of thinking, modelling or acting may serve as examples.

1) "Typical terms" are explained by narratives rather than defined formally. They typically populate social, economic or political domains that widely escape from pure formal descriptions [46]. Lieto and Frixione identify the problem in the formal constraints of description logic underlying the formalism of ontologies. *"As far as typical information is concerned, such formalisms offer only two possibilities: Representing it by resorting to tricks or ad hoc solutions, or, alternatively, ignoring it."* Both, obviously, cannot provide a satisfactory solution.

What is more, the concept of constraints does not work if the problem is not in lacking attributes but in the issue. For example, try to disambiguate the attributes '*motivation'*, '*curiosity*', '*sharp-eyed*', '*commitment*', '*decidedness*' in a formal way (proposed by a senior advisor of a recruitment agency to select chief executives). The separation relies on the narratives, also called "case studies".

The content and particularly the interpretation of narratives, case-studies or attributes are case- and context-sensitive. For example, try to imagine the meaning of the CEO attributes before and after the financial crisis. In the classes, we experience that stories are often differently interpreted. The example about the hedgefonds and Argentina suggests that yet the term 'crisis' can have a negative and a positive connotation.

2) Acceleration, a main topic of this paper, also challenges knowledge management; spill-over learning, is about identifying that knowledge has the potential to "survive" disruptive change. This has already been an issue in high-tech domains. What will become an issue of de-learning when combustion engines of cars are replaced by electrical ones? The about 140 components and related manufacturing techniques, tools, machines, etc. and knowledge may outlast this in the museums or in shops for classic cars. If events of that format happen more frequently, the semantic aspect connected to this is not only to identify perishing and new knowledge. Above all, it is about the paradigm of modelling.

General concepts like 'endurant' (a time-independent observable) and 'perdurant' (only observable over a span of time) or 'universal' (generic term, a pure concept) and 'particular' (subtopic or individual in a generic class) are terms on the highest level of abstraction in ontologies. In case of fast change, the meaning of such concepts may become vague.

3) Is there a semantic support of creativity? In the ARUM project, we made a first approach to construct a simple ontology for the engineering of paper cups based on a matrix of problems to be solved (e.g., subject of innovation), previous solutions (probably outdated) and the attributes, the only category that survived change. As constituents of objects, attributes establish relationships between old and new types of cups but also to further objects, e.g., to the type of coffee shop or drink.

4) This implies that featureless objects (a) don't support relationships, (b) cannot be real in terms of '*perceivable*' and (c) that the perceptiveness and relevance of real objects is in their attributes and the relations they support. This interpretation fits to the definition of knowledge as a network of associations and concepts as the population of these networks.

Formally, "property-is_a-endurant" and "thing-is_a-perdurant" are two statements that hold. Properties also are 'universals' that need instances ('Green' is_constituent_of 'green leaf'). The main point of taking attributes as primary and objects as secondary elements in a conceptual hierarchy is the idea that there are relationships that are not between objects.

This is a radical constructivist approach: objects are constructs of perceived qualities, such as '*green'*. Since attributes are also source of relationships, this approach may pave a path to find new options; there is a potential to augment creativity.

Admittedly far from the topic of this paper, but probably close in terms of semantic modelling, a kind of this problem also appears in physics. The deeper physicists drill into the particularities of the micro-cosmos, the more the properties used to describe an object of interest (electron) dissolve the meaningfulness of these objects and, even beyond, the concept of "object" in general. What is more, an electron is a bundle of properties and the object character of an electron under some conditions renders useless.

"*Today more and more people think that not things are relevant categories but the structures between them. This so-called 'structural realism' is a far more radical break with any conventional atomistic conceptualisation of the material world than any variant of ontologies built on particles and fields*" [47].

This is not the place to discuss structural realism theory. It shall just shed light on the point that turning the ontological pyramid upside down, although in a different context, may be worth a deeper analysis.

VI. KNOWLEDGE MODELS OF RISK LANDSCAPES

A. The Knowledge Base of Managing Unplanned Events

"*An ontology is a specification of a conceptualization.*" with the further explanation: "*A conceptualization is an abstract, simplified view of the world that we wish to represent for some purpose. Every knowledge base, knowledge-based system, or knowledge-level agent is committed to some conceptualization, explicitly or implicitly*" [48]. This section elaborates on the knowledge base about risk related to unplanned events and reduces the scope to *Risk Management* as described in ISO 31000 (2009), to the task of managing negative or positive impacts of events under uncertainty [48].

Time passes in Figure 7 from left to right. The cones represent the universe of past and future events: Given a maximum speed of propagation, events outside delimitations can neither be causes nor effects of the event in the middle.

Those to the left are causes and those to the right effects of the one in the centre. The right cone is the one of management answering to unplanned events. The left cone refers to the responsibility of analysts who, based on history data or simulation, focus on causes of events or paths of propagation and deliver pre-knowledge for the estimation of risk.

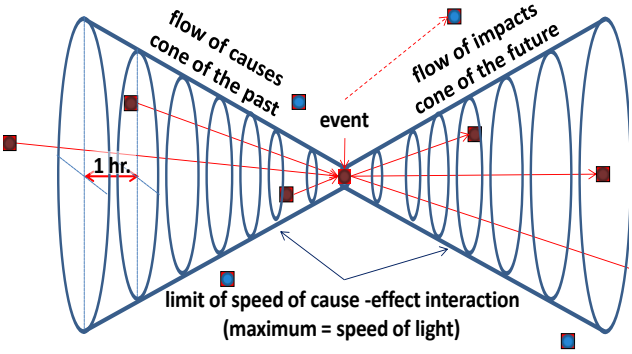


Figure 7. Cone of Cause-Effect Relations and Propagation.

In complex dynamic systems the problem is that it can be very hard, if not impossible, to trace causes or to repeat scenes in simulations. Here we therefore leave analysis to the back-office and focus on the job of managers. Certainly causes matter. But when an unplanned event has arrived, managers are required to appropriately handle a few basic parameters to understand its relevance as well as to act, that is to plan events that recover the situation:

- The Event Risk *(ER)* is equal to a stochastically or statistically defined probability *P*, with $0 \leq P \leq 1$, where 1 and 0 represent certainty of occurrence or non-occurrence.
- The positive or negative (monetary) impact (*I*) of an event is experienced by at least one victim or beneficiary.
- We added the parameter of *awareness* (*A*) to the model as a prerequisite of managerial acting. For example, a competitor's attack may become aware too late. The factor of awareness depends on factors that may antagonize managerial effort, like implicitness, ambiguity, ignorance, taboos, hubris, "*unknowables*" or "*unknown knowables*" [49].
- The *relevance* of events is equal to the *Economic Expectation Value* $eEV = P*I*A$.

Risk landscapes are "*the set of all (possible) events in the managerial universe*" [50] developing from interacting risks. So the value of *P* may be a function of other incidents: the risk of a denial of service attack depends on the probability to hijack a sufficient number of computers. Forward chaining of events is represented as a risk of transit and impact may be mitigated or eliminated by other events (consider noise can-

cellation) or meet a well prepared victim. In turn, a beneficiary of a lottery may not be impressed by the prize.

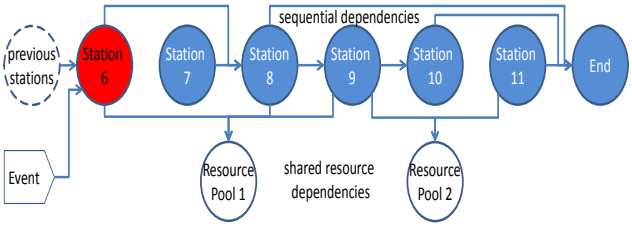


Figure 8. A Simple Pert-diagram of Dependencies.

Unplanned events with a serious impact switch the mode of acting of management from "handle planned operations" into "recover planned status of operations". The background lies in the arguments of the formula $eEV = P*I*A$:

If a substantial economic expectation value would have been identified in previous planning, a proficient management would have planned for that contingency. If *P* or *I* are undervalued, it arrives as unplanned event, and if *A* is close to 0, the event may be an *unknown knowable.*

In the simplest case, RLs connect work-stations along technical dependencies as indicated in Figure 3. But not all stations may be directly connected. A little more complicated model is shown in the Pert diagram [51] in Figure 8: A failure in station 6 may affect station 7 by stopping work in station 8 and shared resources may open another path of propagation. A policy is a plan with the intent to reduce downsides or catch an upside of the related unplanned event, both calculated from the *eEV* of the triggering event with *P*=1 and *A*=1 but still with an impact to be validated.

The evaluation follows the ways of propagation and, therefore, is calculated by the target stations until propagation is stopped. Considering first-level effects only, the impact of an unplanned event in station 6 is equal to the sum of impacts in stations 7 to 11 and consequential idleness of resources. As a process, a policy employs resources and has costs. These resources may be (a) implemental ones like cable-ties that temporarily replace proper brackets for cables, or (b) contingency buffers that may even have been allocated to another purpose. Lifetimes of policies are either limited to the time it needs to find and implement a new or recovering the initial solution, or until a new event asks to change the policy.

B. A Bayesian Model for Estimating Risk

1) The Knowledge Integrator in the Cargo-Lifter project was an experiment in Bayesian inferencing with the ultimate goal to provide investors with a distribution of probabilities about the flow of returns on invest (in terms of capital commitment: the time to amortization) in dependency on technological and organisational progress of work in the project. Simplified, the task is to estimate the likelihood that a dice used in a game is ideal, (analoguously: the project is promising for investors) if it shows in 60 rounds 30 times a 6, i.e., a strongly left-skewed (Figure 9).

The KI, a MAS-based simulator, was used to evaluate alternative strategies for operations' scenarios: What is the economic value of the strategy *OS* for market *OM* and an operations' performance *OP*? Strategies here stand for the managerial options to exploit market potentials, e.g., in terms of the profitability of operations. *OM*, the market environment, was a distribution of market models that included information about volumes or price elasticities, and, particularly, studies with real industrial data. With this high quality of inputs, the distribution of *OM* was considered to be given.

OP, the performance of the airship in this market, was estimated on the base of technological progress regarding functionality offered by the airship (*AF*) and related ground (*GI*) and air infrastructures (*AI*) of airship operations. An example for *GI* is the efforts required to exchange load with industrial locations or with sea and river vessels, and for *AI*, the constraints of an airship operations certificate (issued by air authorities).

If the quality level of the specification of *AF*, *GI* and *AI* is equal to the quality of the market scenarios (*OM*), Bayesian inference would be able to provide a model that answers the questions about returns on investment depending on the progress of the project. But, although the teams improved, there was finally not enough time left to generate a sufficient quality of estimates, because underlying technological knowledge could not be built fast enough to maintain trust.

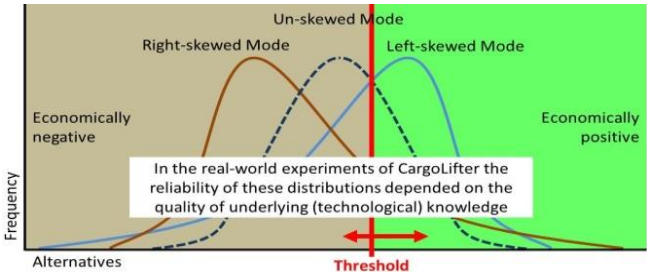


Figure 9. Economic Evaluation of Statistical Modes.

In general, this may be typical for the very early stages of a complex venture and it is not questioned that such a project needs to go through an even painful period of learning. But exactly this needs to build confidence of investors, lead users and the public and not at least self-confidence of the developers about the quality of work to overcome uncertainty: It is to be expected that a traceable path of milestones and lessons learned is a valid strategy to achieve a positive economic expectation value (Figure 9).

This proceeding delivers concrete managerial options [90][91]: If the project performs as expected or even better, a path can be continued and new promising results may justify increasing investment. Reversely, poor or negative progress in overcoming uncertainty, justifies abandoning at least a branch of the path. Finally, if there is a reason to wait for further information, decisions may be postponed for a certain time.

In this early phase of development, the managerial experiments are identical with the number and the distribution of executed managerial options. Thus, although the example of the dice is undercomplex, the analogy holds: A distribution with positive results should have a negative skewness, i.e., the decisions should cluster around a value ≥ 4, a strong 6 may justify increasing investment, and a positive (eco-

nomically negative) skewness of the distribution, i.e., clustering continuously around a value ≤ 3, is a reason to abandon.

The intelligence to turn this into a self-stabilizing process is a core element of the so-called managerial excellence [19]. Executable real (managerial) options are the major tool. In the ADVENTURES project, the respective theoretical model was elaborated by scientists of the Otto Beisheim School of Management (Germany), INSEAD (France), The Wharton School (USA), and in collaboration with the strategic research team of the CargoLifter project [92]. In a sub-project, a first landscape of risk of the venture had been also elaborated.

Real-options models allow translating the problem of a complex venture into the concept of a landscape of risk and the logic of Bayesian inference. It does not mean that any sufficient quality of "priors" is given in the very beginning but that there is a structured proceeding to control the development of that quality and to make reasonable decisions, i.e., to specify thresholds (e.g., minimum progress or maximum failure, Figure 9) and, accordingly, to execute options.

2) Modelling a RL implies learning and a basic quality of data: The examples of the CL-case and of ramp-ups in aviation industry as addressed in the ARUM project illustrate the challenges of building models that are consistent in terms of their data and the semantic model.

In the CL case, it took about two years to get a first, consistent database and a first ontology spanning across the needs of technical engineering as well as of operations and market engineering. In this process we also learned how the development of consistent data depends on a consistent ontology.

The fact that it needed two to achieve this state became an indicator of the weaknesses of the project. Actually it failed because of the unability to structure – within thresholds of time set by the share- and stakeholders – a basic RL for expected outcomes: sufficient returns on invest. Clearly, the capability to learn became the ultimate limitation to the project.

In contrast, the Airbus A350 program does not start from the scratch at all but builds on a large experience of aircraft construction. Nevertheless, as the examples of the introduction of carbon-fibre technology or of lithium-ion batteries show, the inclination of the learning curve is the paramount parameter of success.

Comparing, a simple internet research delivers many examples indicating that 3D-printing will accelerate the speed of innovation in aviation industry (See I-C-7 and [8][9]). Thus, the ontology has to support a frequent and potentially disruptive change.

3) A main aspect of the proceeding is the complexity of the semantic model. This does not imply that all nodes share all their semantics. But collaboration needs a shared core of semantics (Figure 10). It includes the option that a core-ontology of landscapes of risk fits into a larger context. For instance, a core-ontology for production und ramp-up conditions has been elaborated in the ARUM project [80]. Considering that unplanned events are the main issue of risk management, an ontology for events is elaborated in the following section. To maintain maximum compatibility, this ontology is designed according to the standards of the semantic web.

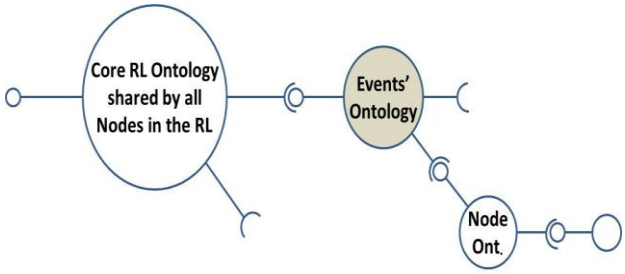


Figure 10. Network of Ontologies.

4) No decision is a decision: Managers, judges and doctors must make decisions, thus, almost inevitably, start from incomplete data and prior knowledge and employ methods that can ground acting on a minimum of plausibility (a state of a best practice, etc.) including complex estimates about coupled and conditioned event risk. They learn from experiment and from (even fatal) failures.

Therefore the ontology needs to support the Bayesian logic of "*inverse probability*" (inferential statistics, a term in early references to Bayes): "*... in practice one can check the dependence on prior distributions by a sensitivity analysis: comparing posterior inferences under different reasonable choices of prior distribution (and, for that matter, different reasonable choices of probability models for data)*" [52].

Section C provides a first model of an ontology, designed according to the W3C standards and aiming to match the requirements of modelling a landscape of risk.

C. A Formal Ontology for Events' Management in an RL

In order to capture the aforementioned concepts, the ontological model shown in Figure 11 was created. The actual event is represented by an individual of the *Event* class and it is linked to the appropriate *EventType* individual through the *hasEventType* property.

The purpose of the *EventType* class is to semantically describe an event. The *triggers* object property enables the expression of the propagation of an *Event*, that is, the occasion when an event causes the triggering of further events. An *Event* is associated with multiple datatype properties.

Namely, the *hasRelevanceValue* property denotes the relevance value of the event, which is compared to the Relevance Threshold (*hasRelevanceThreshold*) in order for responsible roles to decide whether this event has to be handled.

The *hasRisk* datatype property reflects the probability of the event being triggered, whereas the *hasImpactValue* denotes the monetary value of the inflicted impact. Finally, the *raisedAtTime* and *includesComment* properties represent the time the event was raised and any additional comments on the event, respectively. Further datatype properties can be defined and associated with an event, in order to capture all the required information for an event of a specific event type.

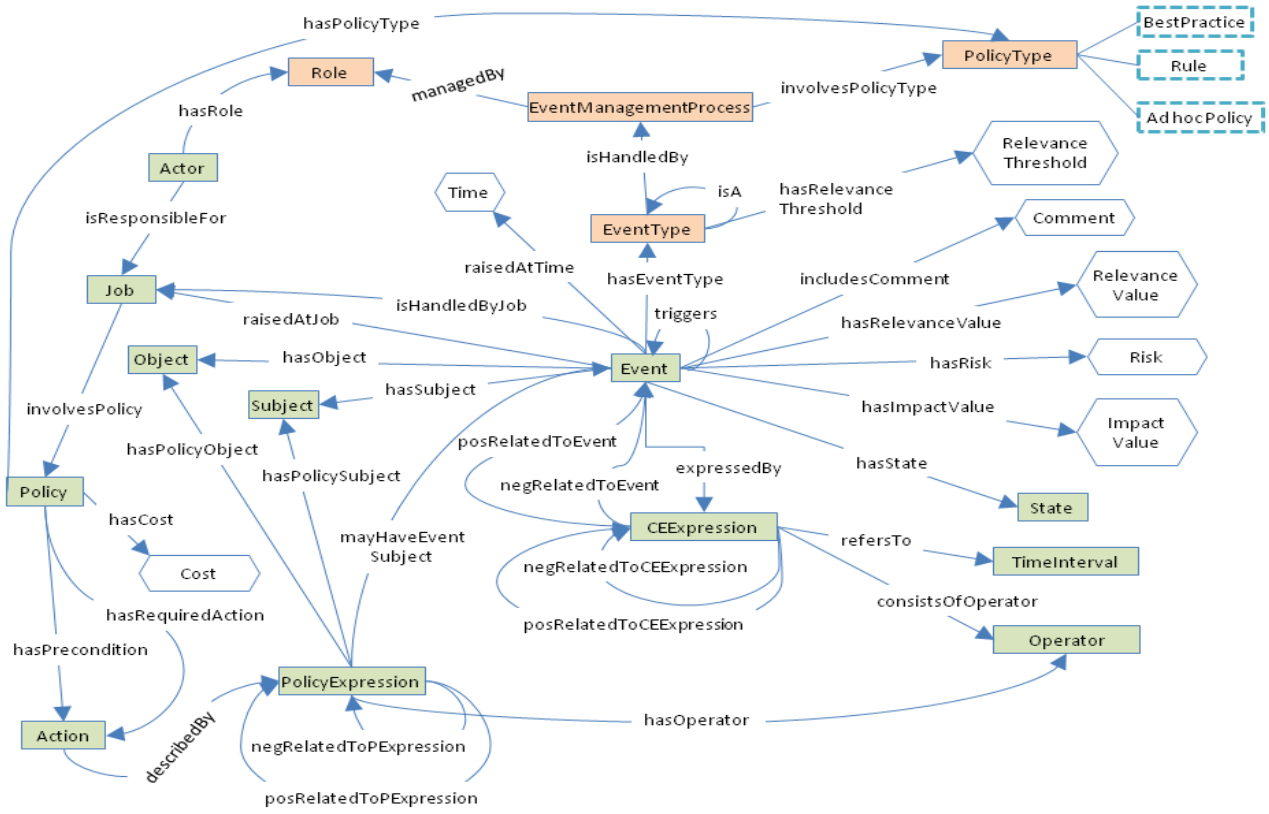


Figure 11. Events Ontology.

An *Event* is associated with the *Subject* class through the *hasSubject* object property, this way expressing the individual that caused the triggering of the event. Apart from a subject, an event may also involve an *Object*, that is, an instance affected by the event. The individuals of the *Object* class are associated to an *Event* via the *hasObject* object property.

The *Job* refers to the atomic task of a scheduled process, during which the event was raised. This association is reflected by the *raisedAtJob* object property.

Through the Job instance, the context of the event can be obtained, following the relations between instances of the class of the Core Ontology that has been developed in the ARUM project [16].

An individual of the *EventManagementProcess* class denotes the process that has to take place in order for the event to be managed effectively. An individual of this class is associated with an individual of the *EventType* class via the *isHandledBy* object property.

The *EventManagementProcess* may serve as a specification of the aforementioned *Job* class. Additionally, the properties *managedBy* and *involvesPolicyType* are defined, linking an Event Management Process to the responsible organisational *Role* and to the appropriate individual(s) of the *PolicyType* class.

The *Role class* is associated with the class EventManagementProcess, depicting the organisational roles that are responsible for the management of an event of a certain type. The specific actor bearing a role is modelled by the *Actor* class, which is associated to the *Role* class through the *hasRole* property. Additionally, the object property *isHandledByJob* is defined, in order to link an *Event* with the scheduled event handling *Job*.

Additionally, an event is associated with an instance of the *State* class through the *hasState* property, denoting if the event is being handled at the moment (Active State) or if it has already been handled (Inactive State).

An Event Management Process involves policies that either mitigate the impact of the event or take advantage of its positive outcome.

An individual of the *Job* class is associated with the *Policy* class through the *involvesPolicy* property, in order to depict the policies that are applied for handling an event. In this case, though, the aforementioned individual of the *Job* class needs to be specified as an Event Handling Job.

A *Policy* can be associated with its *cost* through the datatype property *hasCost* and with a *PolicyType* via the *hasPolicyType* object property.

An individual of the PolicyType class serves as the semantic description of an individual of the Policy class. A PolicyType may fall into three categories:

a. a Best Practice, referring to a policy that has already been documented but not in an identical context, rather than a similar one,
b. a Rule, which reflects a policy that has been applied in the past in an identical situation,
c. an Ad hoc Policy, which is applied in case of no prior knowledge.

The *Action* class is used to express the precondition(s) and the required action(s) that form a *Policy*, in an "if-then" structure. An individual of the *Action* class is linked to an individual of the *Policy* class through the *hasPrecondition* and *hasRequiredAction* properties.

An individual of the *PolicyExpression* class is used to describe an *Action*. Specifically, it is formed by a *Subject* (*hasPolicySubject*) and an *Object* (*hasPolicyObject*), which point to any ARUM Core Ontology class. In the case of the *Subject*, it may also refer to an individual of the *Event* class via the *mayHaveEventSubject* property.

Finally, a *PolicyExpression* is associated with an individual of the *Operation* class through the *hasOperator* property, in order to form a complex Policy Expression. In this case, multiple individuals of the *PolicyExpression* class are linked together via the *posRelatedToPExpression* and *negRelatedToPExpression* object properties.

The purpose of the *CEExpression* class is to enable the modelling of complex events in the form of expressions. The individuals of the *CEExpression* class are associated with an instance of the *Operator* class through the *consistsOfOperator* property, one or more instances of the *Event* class through the *posRelatedToEvent* and *negRelatedToEvent* properties and zero or more individuals of the same class through the *posRelatedToCEExpression* and *negRelatedToCEExpression* properties. For example, considering the atomic events A, B and C, complex events (A AND B) and ((A AND B) OR C) can be modelled.

An instance of the *TimeInterval* class is associated with a *CEExpression* instance via the *refersTo* property, in order to depict the time difference between the triggering of two events, atomic or complex. Individuals of the *Operator* class serve as parts of a *CEExpression* or a *PolicyExpression*. Examples of individuals are the AND, OR, XOR Operators.

D. Ontology Service

Once an event is triggered, decision makers needs to handle it, by exploiting every available piece of information. The role of the Ontology Service is to provide access to the Events Ontology presented in the previous section.

The Ontology service works on two different levels: (a) providing a set of Java libraries to be used as an API to access the attributes of the created objects, as well as their associations with other objects, and (b) as a service, receiving requests from other services in the form of messages. The functionality offered by the Ontology Service is presented in the rest of the section.

One of the responsibilities of the Ontology Service is to provide access to the Events Ontology described in the previous section. The access to the ontological data is required for multiple purposes. Namely, upon the triggering of an event, the actual event, along with information relevant to the context of the event, needs to be stored.

The storage of ontological data is achieved by an internal triple store, a special type of a database, maintaining information in the form of subject-predicate-object triples. It has to be noted that events maintained in the Ontology Service triple store may be delivered by sources maintained by legacy systems, by sensors installed within the factory, introduced by actors in the shop-floor of the factory, etc.

Apart from storing events, the Ontology Service offers the capability of performing queries to the semantic data. The queries can be either applied by invoking the appropriate method offered by the Ontology Service API or it can be expressed in SPARQL [98], which is a query language for RDF. Queries may be performed in the context of retrieving past events with specific characteristics or, in general, events that were raised during a predefined time period, within a specific context.

Based on the semantic type of the event, the Ontology Service is responsible for providing the decision maker with the information regarding whether the event has to be handled or not, by accessing the appropriate value of the relevance threshold and comparing it with the corresponding relevance value, which are reflected in the Events Ontology by means of data type properties.

Then, in the case where the unplanned event needs to be handled, the involved parties have to examine similar events that are logged in the internal triple store, as well as events that were triggered due to the initial event and, finally, infer the probability of them being triggered again. The afore described procedure is accomplished by invoking the appropriate methods of the Ontology Service.

Finally, the Ontology Service offers the functionality of accessing policies, by applying criteria, such as the policy type to be applied on an event of a given type. Furthermore, the most effective policy can be retrieved, by performing comparisons between the attributes of the event under consideration and identical events that were triggered in the past or similar ones that were raised in an identical context. This enables decision makers to select the appropriate policy, based on existing knowledge. If such an event has not been raised again in the past, a new policy can be designed, by following the structure defined by the Events Ontology.

The Ontology Service API makes use of the Apache Jena Framework [53], in order to provide the required functionality of accessing the Events Ontology. Jena is a free, open source Java Framework for building Semantic Web applications and is composed of a number of APIs interacting together to process RDF Data. It fully takes advantage of the RDF data model, by representing semantic information in the form of a graph. This graph is formed by nodes, representing the subject or the objects of a statement, and by edges that are defined by the predicate of a statement.

VII. Computing Landscapes of Operations' Risk

A. The Scale of the Computing Problem

Agent-based modelling and simulation (ABMS) provides means to handle RLs with thousands of distributed nodes as well as means to connect, e.g., a large variety of legacy systems. The choice of algorithms that code a realistic behaviour of agents is a core aspect of modelling. Examples are algorithms to check eligibility of resources to serve in a particular process (workers, tools or components need specified skills) or economic algorithms to minimize idle resources.

Depending on constraints, methods control whether, e.g., agents of components of the product may be active in factory operations or passive in a phase of transfer as one of many

shipments in a ship or as one of many stock-keeping units in a storage. An agent may also represent an item that has become critical due to an event or has been taken under control because of an estimated risk. In the example, each node that represents a workstation, i.e., its share in the breakdown of work (Figure 3) or resources assigned to it, pools of workers, inventories of resources etc. can be represented by one agent, or, if a deeper resolution is required, further agents may represent elements of their substructures (sub-nodes).

In an agent-based model of a risk landscape, the number of nodes is equal to the number of active agents and this number may further change due to the dynamics of the system as events may activate nodes that have been passive before and vice versa. Considering similarities to existing HEC applications, the computational scale of an RL may compare to a weather model with a number of geo-cells equal to the maximum number of nodes of an RL.

While the number of nodes in RLs may be smaller (but dynamically change), the number and variety of interactions is noticeably higher. And while weather models have clear inputs like temperature or humidity, managerial models may have to deal with the question "Is it a problem about humidity?" or with the fact that human behaviour (awareness) may have immediate impact on events' risk and impacts.

However, the computational scale is also driven by response times and needs to specify and compare options to disambiguate interacting events or to identify and implement optimal policies.

Further problems emerge from interactions of operations' domains (Figure 2) if, e.g., one unplanned event drives a lattice tree of potential propagations in production (accruing backlogs) and in parallel in logistic (withdrawals of inventories to avoid quality hazards), or in engineering. Notably, the propagation of an event in a domain can take a "deviation" across another one.

In any case, the processing of complex RL calls for capacity on a level that matches Big V problems, thus hardware and software systems of the classes of High-End and, potentially, High Performance Computing. In this section we describe "candidate" technologies: Cyber-physical systems and applications of Big Data technologies that capitalise on such capacities. Their fit to the management of risk landscapes is obvious, while MAS (as used in the projects described), allow handling of structures and dynamics in any relevant resolution, may lack the required scalability [54].

B. *Cyber-physical Systems (CPS)*

The acronym CPS stands for a widespread integration program embracing various domains from the automation of buildings, car management and communication-based traffic management to factory and supply-chain automation by integrating technological cyber- and real-world (physical) systems into an adaptive operations' automation system.

"*Besides further research ..., a fresh look at CPS also requires a new transdisciplinary engineering approach. As we speak about hybrid systems including electronics, mechanics, software and other technical components, new approaches towards integrated systems modelling approach, a coherent design theory and related design, analysis and simulation tools become indispensable. But, cyber-physical systems are not just a self-contained and isolated ensemble of technical components but are often embedded in a social context to form a socio-technical system. In such systems people are embedded in complex organizational structures and interact with complex infrastructures to perform their work processes. A holistic approach towards human factors, including usability of interfaces and functionality, intuitive machine operating, and seamless coordination of human and machine behaviour are of outmost importance to avoid erroneous system behaviour*" [58].

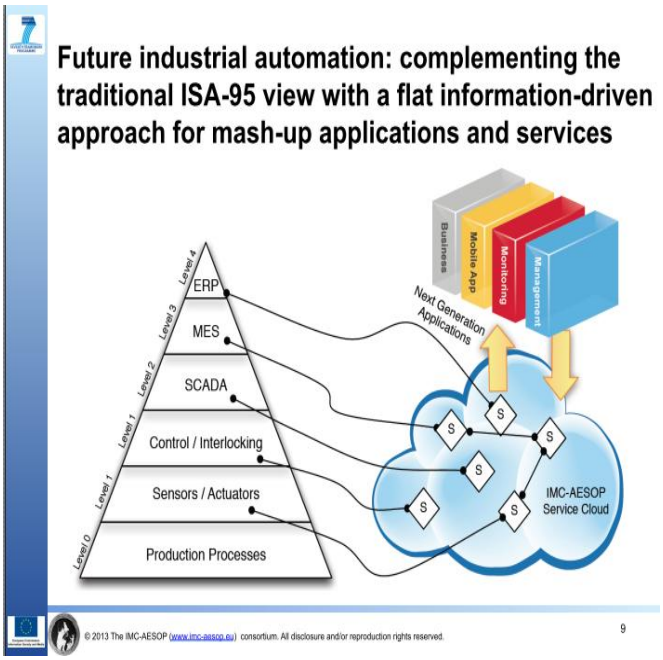


Figure 12. Enterprise Control Hierarchy (left) into a Service Cloud (right) (Factory-case, accordingly to [55] [56] [57])

Figure 12 is adopted from a presentation about the IMC-AESOP project about industrial automation. [55] It depicts a model of a factory application. On the left, it shows an enterprise control hierarchy from low level atomic activity in the workflows up to the overall planning of processes and resources. The hierarchy implies that any instance on a higher level captures and manages many instances of the level below.

The figure shows that CPS integrate the whole control hierarchy of an enterprise from the lowest level of sensors, actors or controllers embedded into robots, machines or attached to materials (RFID), and middleware like System Control and Data Acquisition or Management Execution Systems (SCADA, MES), on the highest level Enterprise Resource Planning Systems (ERP) as services into a "*cyber-system in the cloud*" [60].

An example proves that this infrastructure is compatible to RL-models and the concepts of risk, impact, or unplanned events. On the lowest level of the control hierarchy, an RFID reader may transmit data that a particular "thing" has been read and "now" is "here". On the highest level, an ERP

system may send data about "things" that "now" are expected to be "here", and, if prepared, can provide information about the impact in the case that this plan fails (unplanned event). The compliance to semantic web standards as well as the option to employ semantic technologies allow CPS to cope with heterogeneity of objects and context-dependency of events. Thus, an RL management system can be one of the "next generation applications" mentioned in Figure 12.

But as CPS are restricted to well defined technological objects and relations, the impact of semantic technologies is limited [88], Data generated, stored or processed are available for further processing by Big Data or All-in-Memory applications (e.g., Apache Hadoop [61]), thus not at least for the management of an RL.

C. Big Data Applications

The jungle of data is growing: "*..., there are approximately 4.4 zettabytes (4.4 trillion gigabytes) of information in the digital universe and this number is expected to reach 44 zettabytes by 2020 ... In 2013, by the IDC's* [International Data Corporation] *count, there were 187 billion "connectable things" on the planet, of which 7% were connected to the digital world. By 2020, the number is expected to rise to 212 billion, with 15% ... generating new data*" [60].

Gartner defines Big Data not as a specific architecture or as a particular service but by their scalability to the Big V, namely volume, velocity and variety [61]. Big Data software is a diverse set of technologies and applications, a fast growing crowd of children of the avalanche of data produced by all ways the internet and related technologies are used.

In the Hype Cycle 2013, Big Data is put on the "Peak of Inflated Expectations" (followed by the "Trough of Disillusionment", next to Consumer 3D Printing and ahead of the Internet of Things). The "Plateau of Productivity" is estimated to be reached in 5 – 10 years (while the IoT may take more time to mature). Developments may vary in different markets or due to the size of companies. But the inflation of the data universe is inescapable:

Any organisation that not effectively adopts respective technologies will get a hard time: "*The machine is the problem: A solution is in the machine*" [63].

The ubiquitous production and availability of data in science, in social networks, in business operations, or in commercial or public surveillance systems, etc., actually from any source that could be imagined, fundamentally the digitalisation of almost everything, generates data clouds of zettascale *volume* and high-*velocity* data flows of an unprecedented *variety* and increasing *variability* but also a questionable *veracity*.

Very likely, Big Data applications will develop capabilities that enable the management of very large risk landscapes: The convergence of Big Data and High Performance Computing is standing to reason:

"*The intersection of these two domains is mainly driven by the use of machine learning methodologies to extract knowledge from big data, and we see an increasing number of platforms that are combining these capabilities to provide hybrid environments that can take advantage of data locality to keep the data exchanges over the network at a manageable level while they offer high performance distributed linear algebra libraries*" [64]. A comparison of HPC and Apache Hadoop Big Data architectures is available in [65].

Business intelligence applications are examples that are relevant for RL management. Supply chains or production lines are sources of massive volumes of data. In the Internet of Things, a typical high-resolution and high speed environment also the velocity of data flows is significant. And to a large degree these data are even well structured. Intelligent algorithms can identify patterns in these flows, support the reading of changing patters (e.g., as indicator of impending criticality) or analyse weaknesses in the system or the effectiveness and efficiency of managerial policies and best-practices.

On life-cycle level, patterns can help to isolate sources of ramp-up problems like weaknesses in the design of the product-production systems, ineffective practices, etc. The analysis of communication flows in social media can provide early indication of a changing customer behaviour, like variations in the trend towards a shareconomy.

Global business platforms are a new development, e.g., collecting massive social data consumer and businesses in the first instance for improving and diversifying their own business. Finally they may become also providers of business intelligence services.

The inflation of data also stands for growing resolution and variety, both implying more sources or targets of risk as well as for acceleration (see Figure 5). The closer to real-time, the less volume and the more velocity and variety will be the core of the task to specify, find, prepare and process the *right* data to support the management of risk landscapes timely (with respect to velocity) and properly (regarding to variety, variability and veracity).

Data-intelligence will have a significant potential to improve business and operations' intelligence in the management of risk landscapes. They identify early problem indicators, analyse the vulnerability of business- and of operations' systems, track interactions between degrees of freedom in dimensions of operations, scanning and analysing drivers of change or positive feedbacks.

The job of data scientists is to develop time-effective semantic models and algorithms for mining meaningful data and asking meaningful questions for further processing and technical support of decision making in the light of valid strategies, finally for the purpose to support R.E.A.L. processes in strategic and operations' management. Beyond, substantial capabilities of "creative criticism" will become paramount: In digitalised, fast changing, heterogeneous, and potentially compromised environments the value is rather in the questions than in the answers.

For illustration we borrow a case from the financial market: Sketchily the financial crisis may have been caused by under-complex models or algorithms respectively by highly leveraged lending that had not been questioned. On a deeper level the reasons are in unquestioned institutional failures that blocked the emergency exits: "*As long as the music is playing, you've got to get up and dance ...*" (Charles Prince, Chairman and CEO of Citigroup Inc. after an about 80 billion USD loss in assets) [67].

Another possible trap may lie in community effects that trigger positive feedbacks and risk. For example, unquestioned predictions in stock markets can become self-fulfilling prophecies, like outcomes of search engines depends on hidden rankings by the search engine operator.

D. Multi-agent Systems (MAS)

MAS are no to CPS or BD-applications but may rather be enablers for intelligent simulation, planning and scheduling as well as to cope with the dynamics and the resolution of detail of RL. An example may be a local unplanned event driving the global criticality of a particular resource that, to mitigate the problem, now need particular care by activating a respective software agent. Given the computational scale of RL the problems may be the scalability of MAS and the need to control the typically high loads of communication MAS impose to the infrastructure.

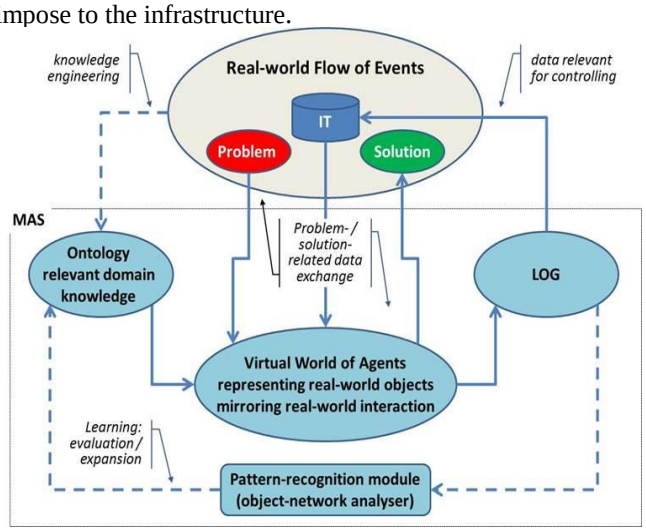


Figure 13. Architecture of an MAS mimicking a real in a virtual world [68].

1) The CargoLifter Knowledge Integrator and CESSAR were realised as multi-agent systems that mimic a relevant part of the world on the base of an ontology. Software agents were applied to objects representing a relevant risk (e.g., a failure would interrupt operations). Relations of agents were designed accordingly to a service-driven logic, analogously but not in full conformity to standards of a service-oriented software architecture [86].

Multi-agent systems (MAS) [86][87] consist of software agents that collaborate in swarms to pursue common, while each pursuing its individual goals. As semantic agents they share a set of concepts (the ontology) that enables to reason about scenes in a compatible way and to coordinate action so that local activity of agents becomes effective in terms of global goal. Semantic agents may be designed for learning from outcomes of decisions, e.g., by comparing current and previous scenes. In the CL or the CESSAR model, agents pursued individual goals on a virtual market by providing services and procuring services needed for their business. Both were stand-alone systems that may exchange data via Internet but not based on Web technology standards. Figure 13 depicts the architecture of these models.

2) There are advantages of this approach: Service-oriented modelling in general is intuitive and the architecture of agent systems allows to design highly complex and multi-layered service supply chains or to integrate any relevant resolution of detail are further advantages of these systems:

- The set of services can be increased by adding agents with respective demands and offering capabilities (truck transport + cleaning + maintenance ...)
- The volume of services is controlled by constraints of capacity only (availability of trucks for service).
- The resolution of object can be increased by adding agents as providers of sub-services via is_part_of relations and service dependencies (in the iCRFID project, e.g., truck → tank → sensor → gas station → pump_#)
- Organising of managerial tasks by multiple holons (e.g., linking all fuel-sensors into a system wide fuel-management *and* into a truck maintenance system).
- True 1:1 interactions between classes (ARUM project: sensor ↔ station) or individual agents (iCRFID project: good_X ↔ passenger_Y) and between events can be implemented as 'objects'.
- The resolution of time is equal to the number of events / unit of time (frequency) and depends on response-times of the MAS to unplanned events.
- Rationales of acting and best practices are implemented in the economics of agents (e.g., transaction costs depending on degree of propagation).
- Systematic exploitation of discretions to act (DTA / slack) by iterative negotiations [84].

3) MAS have not joined the mainstream: *"Despite considerable progress, it seems that the challenges ... encountered at early days still hold. In particular, the adoption of AOSE* [Agent-Oriented Software Engineering] *principles in the academia, and even more so in the industry, is limited*" [69].

Besides the lack of scalability, this is likely also owed to the fact that MAS are able to solve complex problems because they are complex systems: *Solutions are sensitive to start conditions and non reducible to the behaviour of individual agents, thus emergent and hardly reproducible*. From practical experience as managers or engineers, who typically claim to *control* systems, we know that MAS quite plainly prove that this claim mostly is an illusion: "*It works, but I don't know why*." (an experienced dispatcher at Cologne Airport after experiments with *CESSAR* (iCRFID project)).

The scale of RLs require MAS to be redesigned for running on larger and likely also parallel HEC computing infrastructures [70]. Also scalability is still an issue, although holonic MAS architectures enable larger system [71][72]. But also this strategy will hardly solve the problem for very large loads of communication. For instance, a full scale industrial version of *CESSAR* (iCRFID project) where any unplanned event can drive impact across 18,500 flights, operated by 4,460 aircrafts that connect 1,860 destinations for almost 17 Million passengers per day [73].

The development of hybrid MAS is another strategy to reduce the load of communication by employing highly performant mathematical solvers for the calculatory jobs and

semantic agents to structure scenes and problems as well as to evaluate results or to support related learning processes.

4) Architectures of CPS and MAS have common features: MAS consist of service-oriented autonomous agents interacting in a cyberspace accordingly to FIPA standards (Foundation for Intelligent Physical Agents) [74][88]. CPS consist of service oriented *applications* (embedded software included) that interact accordingly to web-service standards in the Internet of Services. But while an MAS is a self organising swarm of autonomous, goal-driven agents, a CPS rather compares to an orchestra expected to deliver a particular performance that has been composed by management: the plan of operations, e.g., of a factory or of a complex network of supply chains. Nevertheless the "orchestra" should be able of changing music on demand, e.g., of an unplanned event.

In terms of the R.E.A.L. framework, the equivalent to an alternative piece of music is an alternative plan, either in the form of defined rules or, more complex, as elaborated best practice. MAS provide more flexibility to extend and adapt models or create ad-hoc peer-to-peer relationships between agents respectively actors. And finally MAS are able to simulate and analyse complex systems while a CPS is an architecture that as such needs to employ integrated services, e.g., a MAS for simulation.

5) Big Data may include MAS-based services, e.g., for dynamic planning and scheduling, for simulation and analysis or as semantic reasoners. MAS on the level of high-end or high-performance computing are in reach of development [70], particularly for offline computing task (i.e., not as real-time controllers in a CPS). Promising approaches may be to use BD applications to feed into MAS or, vice versa, to enable Big Data by MAS, e.g., for experiments in risk management for strategic scenarios: "*Who may consider this information to be valuable? … "What would happen if we provide our product or service free of charge? What if a competitor did so? The responses should provide indications of the opportunities for disruption, as well as of vulnerabilities*" [75]. In operations, 1:1-designed scenarios can be simulated like *"What is the impact of an organisational change to vulnerability?" "What are the limits of current best practices to mitigate impacts of a particular class of unplanned events?"*

E. Limitations to Effectively Parallelizing MAS

MAS are generic distributed systems. This may suggest that agents in a MAS act in parallel. But this is not true. Most MAS are deployed on Microsoft standard software and agents' decision making and communications are sequentially scheduled by allocating capacity slots to tasks or threads. If parallel processors are available and supported by the operations system typically tasks can be distributed. Also holonic architectures can be processed in parallel [76].

The variety of agents' operations systems that allow for real parallel acting of MAS is very limited. Problems lie in the internal communication of agents. Besides the volume of data traffic the messaging protocols of agents are hardly compatible with operations' systems like MPI that are used in parallel computing.

In the context of our work the Repast HPC platform has been analysed [56]. This technology supports parallel agents' activity in an HPC environment, supports large models and enables the communication between agents. However Repast is based on an internal time model the platform is unable to continuously exchange information and synchronize with external systems in real-time / real world. It does not support scheduling or dynamic planning of ongoing operations, that is "online" with actual processes. In consequence Repast is no tool that can be integrated into a service cloud and its use is restricted to simulation.

F. HEC computing architectures and scenarios

With many modern and often dynamical and interactive application scenarios, the term "high performance" is covering demanding applications that are on the one hand compute- and on the other hand data-centric. It is a common understanding that parts of the respective scenarios will support the exploitation of parallelism for their implementation.

With all available high end and high performance systems and architectures the hardware and software issues cannot be separated. The requirements from algorithms and application scenarios lead to solutions favouring the different architectures. In the case of increasingly big data scenarios the attributes of the data and usage are a most important factor.

With the workflows and algorithms the most major attributes of the data, namely volume, velocity, variability, and vitality, mark the physical requirements of needs for communication and data locality. The respective software components have to be adapted in order to fit these requirements, which have to span from distributed to centralised resources, creating robust, reliable, and intelligent software components and workflows.

High End Computing (HEC) systems range from a desktop computer, through clusters of servers and data centres up to high-end custom supercomputers. Resources can be physically close to each other, e.g., in a highly performant compute systems, or the compute power can be distributed on a large number of computers as with most Grid and Cloud computing concepts. Mostly, these architectures are used for task-parallel and data-parallel problems in classical capacity computing.

High Performance Computing (HPC) systems are based on architectures with a large number of processors, for exploiting massive parallelism. Commonly used models are Massively Parallel Processing and Symmetric Multi-Processing, used with the concept of local islands. Due to physically shared memory usage and compute communication, the physical architectures with these HPC systems are different.

Handling of RM processes will therefore focus on distributed components. Due to the physically different structure of highly distributed and massively parallel resources, the following aspects can be considered.

In the case of HEC, e.g., Cloud Computing, these components can be system resources acting autonomously like servers, being connected by external network means, being the ideal resources for events processing at capacity level. HEC resources can provide efficient means for massively

distributed tasks. The non-availability of resources can be handled on a job or task base.

In the case of HPC, e.g., common with Scientific Computing on Supercomputing resources, the components can be internal network resources only, compute nodes on the one hand, being controlled by a management network and software, and management nodes on the other hand.

The communication intensive modelling especially for the overall results and visualisation as well as the pre- and post-processing for the models will be suitable for use of HPC resources. In order to optimise the efficiency and economic use of the HPC resources and minimising the effects of job size fragmentation these resources should be used for a defined class of suitable large tasks within the workflow. Available resources can be configured as distributed HPC resources within the network provided for the described systems. Regarding the demanding network requirements Software Defined Networks (SDN) [77] can provide modular and efficient solutions for these purposes.

VIII. Conclusions and Further Work

In most countries, listed firms are required to include a formal analysis of corporate risk in annual reports. Theoretical and descriptive parts are delivered in narrative form. The standard of underlying risk models is based on actuarial methodology that also may deal with relevant operations' risk.

They provide an integrating system of strategies, similar to those applied in insurance business and with similar problems as discussed in this paper: "*A major challenge here is a more substantial and realistic description and modelling of the various complex dependence structures between risks that show up on all scales*" [78]. But integrated risk modelling and processing, as addressed in this paper, is far too detailed and complex to be by this rather formal approach.

Although our work is in an early stadium, the industrial use-cases provide confidence that the particular computational approach discussed above will add a new strategy to risk management under exceptional circumstances in real economy. For operation and management, it is appropriate to focus on risk landscapes as networks of nodes and of related service levels [79][80]. Therefore, events described and related processes can be handled with less interference if services are defined and interfaces for the processes are created.

This is important for the HEC, HPC, and communication resources required. For HEC processes, this can be done on a service level cloud base, whereas for the HPC resources available in research environments, this mostly will have to be assisted by service level agreement policies.

This is important for the HEC, HPC, and communication resources required. For HEC processes, this can be done on a service level cloud base, whereas for the HPC resources available in research environments, this mostly will have to be assisted by service level agreement policies.

In both fields of semantic modelling and computation of industrial landscapes of risk, further work is to be done. The most crucial issues are

- To elaborate a formalised architecture of RL, based on the network of nodes, but consistently including the large variety of structural and dynamic aspects on the required level of detail.
- To develop an effective Bayesian strategy of capturing and improving estimates of event risk and related impact from responsible managers. The issue is that hybrid models require to link semantic conceptualisation with Bayesian methodology [38] that significantly goes beyond the eEV-model used in this paper. Another aspect is that relations between ontological and process-based reasoning (things and flows) may have to be revised [41].
- To deliver a first concrete industrial model of a risk landscape.

Acknowledgments

This work is partially supported by the EU FP7 Programme, ARUM project, GA- No. 31405.

References

[1] U. Inden, D. T. Meridou, M.-E. Ch. Papadopoulou, A.-C. G. Anadiotis, and C.-P. Rückemann, "Complex Landscapes of Risk in Operations Systems Aspects of Processing and Modelling," The Third International Conference on Advanced Communications and Computation (INFOCOMP 2013) IARIA, Nov. 2013, pp. 99-104, ISSN: 2308-3484, ISBN: 978-1-61208-310-0.

[2] A. Dixit and R. Pindyck, "Investment Under Uncertainty", Princeton University Press, 1994.

[3] H. Bruch and S. Goshal, "Management is the Art of Doing and Getting Done," Business Strategy Review, Sept. 2004, vol. 15, pp.4-13.

[4] K. Ishikawa, Introduction to Quality Control, Productivity Press, 1990.

[5] E.-K. Boukas and R. P. Malhame, "Analysis, Control and Optimization of Complex Dynamic Systems," Springer, 2005.

[6] A. M. Brandenburger and B. J. Nalebuff, "Co-Opetition," Crown Business, 1996.

[7] N. N. Talib, "The Black Swan: The Impact of the Highly Improbable (2nd Edition)," Random House, May 2010, ISBN: 978-1400063512.

[8] Why Project Fails. *Airbus – A380*. [Online]. Available from: http://calleam.com/WTPF/?p=4700/ 2014.11.14

[9] Why Project Fails. *Boeing Commercial Aeroplanes*. [Online]. Available from: http://calleam.com/WTPF/?p=4617 2014.11.14

[10] F. Krings and O. Krone, "Konzept zur interactiven Transportsteuerung im Rahmen globaler Verkehrssysteme," Technische Universität Carolo Wilhelmina Braunschweig, 1996.

[11] R. Kohlenberg and R. Schulze, "Interoperabilität von verschiedenen Barcodesystemen zur Sendungsverfolgung in der Logistikkette VW – VWdM," Technische Universität Carolo Wilhelmina Braunschweig, 1997.

[12] GLORI. *Global Logistics Research Initiative*. [Online]. Available from: www.glori.com/research.htm/ 2014.11.14

[13] Bundesministerium für Wirtschaft und Energie. *iC-RFID: RFID-gestützte Servicesysteme*. [Online]. Available from: http://bmwi.de/DE/Themen/Digitale-Welt/Internet-der-Zukunft/internet-der-dinge,did=360478.html/ 2014.11.14

[14] aerospace-technology.com. *CargoLifter CL160*. [Online]. Available from: www.aerospace-technology.com/projects/cargolifter/ 2014.11.14

[15] R. Franken and U. Inden, "CESSAR – Configuration and Evaluation of Service Systems in Air-Catering with RFID," Cologne Business Working Papers, Dec. 2011. ISSN: 2192-936.

[16] ARUM. *Adaptive Production Management*. [Online]. Available: http://www.arum-project.eu/ 2014.11.14

[17] S. L. Goldman, R. N. Nagel, and K. Preiss, "Agile Competitors and Virtual Organisations: Strategies for Enriching the Customer," Wiley, 1995.

[18] K. Preiss, S. L. Goldman, and R. N. Nagel, "Cooperate to Compete: Building Agile Business Relationships," Wiley, 1996

[19] Industrial Excellence Award. *Benchmarking Management Quality for European Competitiveness*. [Online]. Available from: http://de.industrial-excellence-award.eu/home/ 2014.11.14

[20] P. Hermanns, "Organizational Hubris – Aufstieg und Fall einer Celebrity Firm am Beispiel der CargoLifter AG", Kölner Wissenschaftsverlag, 2012. ISBN 978-3-942720-33-5.

[21] R. Bea. *Approaches to achieve adequate quality and reliability*. [Online]. Available from: http://ccrm.berkeley.edu/pdfs_papers/bea_pdfs/quality-and-reliability.pdf 2014.11.14

[22] S. A. Srinivasa Moorthy, "Lifecycle Challenges in Long Life and Regulated Industry Products," ICoRD'13, Lecture Notes in Mechanical Engineering, Springer India, pp. 833-844, 2013, doi:10.1007/978-81-322-1050-4_66.

[23] A. McAfee and E. Brynjolfsson, "Big Data: The Management Revolution," Harvard Business Review, pp. 61-68, Oct. 2012.

[24] Investopedia. *Working Capital*. [Online]. Available from: http://www.investopedia.com/terms/w/workingcapital.asp 2014.11.14

[25] W. Ashby, "The Law of Requisite Variety," An Introduction to Cybernetics, Chapman & Hall, 1956, pp. 206–212, ISBN:0-416-68300-2.

[26] W. Ashby. *The W. Ross Ashby Digital Archive*. [Online]. Available from: http://www.rossashby.info/journal/page/4158.html 2014.11.14

[27] U. Inden, G. Lioudakis, and C.-P. Rückemann, "Awareness-Based Security Management for Complex and Internet-Based Operations Management Systems," Integrated Information and Computing Systems for Natural, Spatial, and Social Sciences, IGI Global, 2013, pp. 43-73, ISBN 978-1-4666-2190-9.

[28] SKYbrary. *Airborne Separation Assurance Systems (ASAS)*. [Online]. Available from: http://www.skybrary.aero/index.php/Airborne_Separation_Assurance_Systems_%28ASAS%29 2014.11.14

[29] R. Kurzweil. *The Law of Accelerating Returns*. [Online]. Available from: http://www.kurzweilai.net/the-law-of-accelerating-returns 2014.11.14

[30] J. A. Schumpeter, "Capitalism, Socialism and Democracy". Taylor & Francis e-Library, 2003, ISBN 0-415-10762-8.

[31] Reuters Deutschland. *Osram-Mitarbeitern stehen Kündigungen ins Haus*. [Online]. Available from: http://de.reuters.com/article/companiesNews/idDEKBN0FZ1H120140730 2014.11.14

[32] C. Christensen, "Das Dilemma der Kapitalisten," Harvard Business Manager, 2014.

[33] T. Berners Lee, J. Hendler, and O. Lassila. *The Semantic Web*. [Online]. Available from: http://www.scientificamerican.com/article/the-semantic-web/ 2014.11.14

[34] The University of Sydney. *A step closer to bio-printing transplantable tissues and organs*. [Online]. Available from: http://sydney.edu.au/news/84.html?newsstoryid=13715 2014.11.14

[35] U.S. Army. *Chow from a 3-D printer? Natick researchers are working on it*. [Online]. Available from: http://www.army.mil/article/130154/Chow_from_a_3_D_printer__Natick_researchers_are_working_on_it/ 2014.11.14

[36] 3D Print Canal House. *What is a 3D Print Canal House?*. [Online]. Available from: http://3dprintcanalhouse.com/what-is-the-3d-print-canal-house-2 2014.11.14

[37] Engineering & Technology magazine (E&T). *Metal 3D printing promises revolution in aerospace*. [Online]. Available from: http://eandt.theiet.org/news/2013/oct/metal-3d-printing.cfm 2014.11.14

[38] Travel Daily Asia. *Airbus plans to make aircraft using 3D printers.* [Online]. Available from: http://www.traveldailymedia.com/205079/airbus-plans-to-make-aircraft-using-3d-printers/ 2014.11.14

[39] 3D Printer. *GE Announces Production 3D Printing and Stock Goes Up*. [Online]. Available from: www.3dprinter.net/ge-announces-production-3d-printing-stock-goes 2014.11.14

[40] 3D Printer. *British RAF Fighter Jets Fly with 3D Printed Parts for the First Time*. [Online]. Available from: http://www.3dprinter.net/british-raf-fighter-jets-fly-3d-printed-parts-first-time 2014.11.16

[41] A. Reichental. *How 3D printing will turn us all (back) into makers: Avi Reichental at TED2014*. [Online]. Available from: http://blog.ted.com/2014/03/19/how-3d-printing-will-turn-us-all-back-into-makers-avi-reichental-at-ted2014/ 2014.11.16

[42] D. Etherington. *Amazon Launches A 3D Printing Store With Customizable Goods*. [Online]. Available from: http://techcrunch.com/2014/07/28/amazon-launches-a-3d-printing-store-with-customizable-goods/ 2014.11.16

[43] EXPLORA Frankfurt Science Centre. *Museum + Wissen + Tech + Arts*. [Online]. Available from: http://www.explora.info/pressepix/pressepix9.php 2014.11.16

[44] K. E. Ch. Asch, "A Very Practical Geoinformatics Project: The Reality of Delivering a Harmonized Pan-European Spatial Geoscience Database," Geoinformatics 2007 - Data to Knowledge, GSA, May 2007, pp. 4-5.

[45] W. Kuhn, "Semantic engineering," Research Trends in Geographic Information Science, Springer, pp. 63–76, Jun. 2009, doi. 10.1007/978-3-540-88244-2_5, ISSN: 1863-2246, ISBN: 978-3-540-88244-2.

[46] M. Frixione and A. Lieto, "Towards an Extended Model of Conceptual Representations in Formal Ontologies: A Typicality-Based Proposal," Journal of Universal Computer Science, vol. 20(3), pp. 257-276, Mar. 2014.

[47] M. Kuhlmann. *Was ist real?*. [Online]. Available from: http://www.spektrum.de/alias/titelthema-quantenfeldtheorie/was-ist-real/1286309 2014.11.16

[48] ISO Store. *ISO 31000:2009, Risk management - Principles and guidelines.* [Online]. Available from: http://www.iso.org/iso/catalogue_detail?csnumber=43170 2014.11.16

[49] N. A. Doherty and A. Muermann, "On the Role of Insurance Brokers in Resolving the Known, the Unknown and the Unknowable," The Known, the Unknown, and the Unknowable in Financial Risk Management: Measurement and Theory

Advancing Practice, Princeton University Press, pp. 194-209, 2010.

[50] J. L. Synge. *Events and Spacetime* [Online]. Available from: http://www.phy.syr.edu/courses/modules/LIGHTCONE/events.html 2014.11.16

[51] J. A. Rooke, L. J. Koskela, and D. Seymour, "Producing things or production flows? Ontological assumptions on the thinking of managers and professionals in construction," Journal of Construction Management and Economics, Taylor & Francis, vol. 25 (10), pp. 1077-1085, Oct. 2007, ISSN: 0144-6193.

[52] A. Gelman, "Prior distribution," Encyclopedia of Environmetrics, Wiley, vol. 3, pp. 1634-1637, 2002, ISBN: 0471 899976.

[53] The Apache Software Foundation. *Apache Jena*. [Online]. Available from: http://jena.apache.org/index.html/ 2014.11.17

[54] P. Leitão, U. Inden, and C.-P. Rückemann, "Parallelizing Multi-gent Systems for High Performance Computing," The Third International Conference on Advanced Communications and Computation (INFOCOMP 2013) IARIA, Nov. 2013, pp. 1-6, ISSN: 2308-3484, ISBN: 978-1-61208-310-0.

[55] S. Karnouskos, "Cloud-based Cyber-Physical Systems in Industrial Automation," The 39th Conference of the IEEE Industrial Electronics Society (IECON 2013), IEEE, Nov. 2013.

[56] The IMC-AESOP Consortium. *IMC-AESOP Project: Architecture for Service Oriented Process.* [Online]. Available from: http://www.imc-aesop.eu/dl/13-4285%20AESOP%20broschyr_skiss4.pdf 2014.11.16

[57] A. W. Colombo, S. Karnouskos, and T. Bangemann, "A System of systems view on collaborative industrial automation," IEEE International Conference on Industrial Technology (ICIT 2013), IEEE, Feb. 2013, pp. 1968 – 1975, ISBN: 978-1-4673-4568-2, doi: 10.1109/ICIT.2013.6505980.

[58] B. J. Krämer, "Evolution of Cyber-Physical Systems: A Brief Review," Applied Cyber-Physical Systems, Springer New York, 2014, pp. 1-3, ISBN: 978-1-4614-7335-0, doi: 10.1007/978-1-4614-7336-7_1.

[59] Apache Software Foundation, "Apache Hadoop," [Online]. Available from: http://hadoop.apache.org / [Last accessed 16/11/2014].

[60] A. Gibson. *Growth of Big Data a big challenge for business*. [Online]. Available from: http://www.strategic-risk-global.com/growth-of-big-data-a-big-challenge-for-business/1408171.article 2014.11.16

[61] Gartner IT Glossary, "Big Data," [Online]. Available from: http://www.gartner.com/it-glossary/big-data/ [Last accessed 16/11/2014].

[62] N. Heudecker. *Hype Cycle for Big Data, 2013*. [Online]. Available from: https://www.gartner.com/doc/2574616 2014.11.16

[63] Y. Poullet, "EU data protection policy. The Directive 95/46/EC: Ten years after," Computer Law and Security Review, Elsevier, 2006, vol. 22, Issue 3, pp. 106-127, doi: 10.1016/j.clsr.2006.03.004.

[64] F. Villanustre. *Big Data Technologies Narrow the Gap between HPC and the Enterprise*. [Online]. Available from: http://www.isc-events.com/bigdata13/pressreleases-reader/items/big-data-technologies-narrow-the-gap-between-hpc-and-the-enterprise.html 2014.11.16

[65] S. Jha, J. Qiu, A. Luckow, P. Mantha, and G. C. Fox. *A Tale of Two Data-Intensive Paradigms: Applications, Abstractions, and Architectures*. [Online]. Available from: http://arxiv.org/abs/1403.1528 2014.11.17

[66] H. Adams and P. Varadan. *Fighting Financial Crime With Data.* [Online]. Available from: http://www.accenture.com/SiteCollectionDocuments/PDF/Accenture-Fighting-Financial-Crime-with-Data.pdf 2014.11.17

[67] Reuters. *Ex-Citi CEO defends dancing quote to U.S. panel*. [Online]. Available from: www.reuters.com/assets/print?aid=USN0819810820100408 2014.11.17

[68] G. Rzevski, "A practical Methodology for Managing Complexity," Emergence: Complexity and Organization–an International Transdisciplinary Journal of Complex Social Systems, vol.13, pp. 38-56, 2011.

[69] A. Sturm and O. Shehory, "Agent-Oriented Software Engineering: Revisiting the State of the Art," Agent-Oriented Software Engineering, Springer Berling Heidelberg, 2014, pp 13-26, ISBN: 978-3-642-54432-3, doi: 10.1007/978-3-642-54432-3_2

[70] P. Leitão, U. Inden, and C.-P. Rückemann, "Case Studies for Parallelising Multi-Agent Systems for High Performance Computing," International Journal On Advances in Software, IARIA, 2015 (to appear)

[71] H. Van Brussel, J. Wyns, P. Valckenaers, L. Bongaerts, and P. Peeters, "Reference Architecture for Holonic Manufacturing Systems: PROSA," Computers in Industry, Elsevier, Nov. 1998, vol. 37, Issue 3, pp. 225-276, doi:10.1016/S0166-3615(98)00102-X.

[72] A. Koestler, "The Ghost in the Machine," Hutchinson & Co, 1967.

[73] Star Alliance. *Travel the World with the Star Alliance Network.* [Online]. Available from: http://www.staralliance.com/de/about/member_airlines/ 2014.11.17

[74] IEEE computer society. *Foundation for Intelligent Physical Agents (FIPA)*. [Online]. Available from: http://www.computer.org/portal/web/sab/foundation-intelligent-physical-agents/ 2014.11.17

[75] J. Bughin, M. Chui, and J. Manyika. *Clouds, big data, and smart assets: Ten tech-enabled business trends to watch*. [Online]. Available from: http://www.mckinsey.com/insights/high_tech_telecoms_internet/clouds_big_data_and_smart_assets_ten_tech-enabled_business_trends_to_watch/ 2014.11.17

[76] P. Leitão and J. Barbosa, "Adaptive Scheduling based on Self-organized Holonic Swarm of Schedulers," IEEE 23rd International Symposium on Industrial Electronics (ISIE), IEEE, Jun. 2014, pp. 1706-1711, doi: 10.1109/ISIE.2014.6864872.

[77] A. Georgi, R. Budich, Y. Meeres, R. Sperber, and H. Hérenger, "An Integrated SDN Architecture for Applications Relying on Huge, Geographically Dispersed Datasets," The Third International Conference on Advanced Communications and Computation (INFOCOMP 2013) IARIA, Nov. 2013, pp. 129-134, ISSN: 2308-3484, ISBN: 978-1-61208-310-0.

[78] D. Pfeifer and J. Nešlehová, "Modelling and generating dependent risk processes for IRM and DFA," ASTIN Bulletin, Peeters, 2004, vol. 34, pp. 333-360, doi: 10.2143/AST.34.2.505147.

[79] C.-P. Rückemann, "Enabling Dynamical Use of Integrated Systems and Scientific Supercomputing. Resources for Archaeological Information Systems," The Second International Conference on Advanced Communications and Computation (INFOCOMP 2012) IARIA, Oct. 2012, pp. 36-41, ISBN: 978-953-307-737-6.

[80] C.-P. Rückemann, "Queueing Aspects of Integrated Information and Computing Systems in Geosciences and Natural Sciences," Advances in Data, Methods, Models and Their Applications in Geoscience, InTech, Dec. 2011, pp. 1-26, ISBN: 978-953-307-737-6, doi: 10.5772/29337.

[81] J. Ríos, F. Mas, and J. L. Menéndez, "Aircraft Final Assembly Line Balancing and Workload Smoothing: A Methodical Analysis," Kay Engineering Materials, Trans Tech Publications, Feb. 2012, vol. 502, pp. 19–24, ISSN: 1662-9795, doi: 10.4028/www.scientific.net/KEM.502.

[82] K. Sundaram and F. Tomesco. *Boeing Disputes India Report of $500 Million 787 Payment*. [Online]. Available from: http://www.businessweek.com/news/2012-03-14/boeing-to-pay-air-india-500-million-on-787-delays-india-says/ 2014.11.17

[83] E. Fleisch and G. Müller-Stewens, "High-Resolution-Management: Konsequenzen des Internet der Dinge auf die Unternehmensführung," Zeitschrift Führung + Organisation (ZfO), Schäffer-Poeschel, 2008, vol. 77, pp. 272 - 281, ISSN 0722-7485.

[84] U. Inden, S. Naimark, and C.-P. Rückemann, "Towards a Discretion-to-Act Control Architecture by Decoupling Modelling from Complexity," TMC Academic Journal, TMC Academy, Feb. 2013, vol. 7, ISSN: 1793-6020.

[85] L. Feigenbaum. *Semantic Web vs. Semantic Technologies.* [Online]. Available from: http://www.cambridgesemantics.com/de/semantic-university/semantic-web-vs-semantic-technologies/ 2014.11.17

[86] P. Skobelev, "Bio-Inspired Multi-Agent Technology for Industrial Applications," Multi-Agent Systems - Modeling, Control, Programming, Simulations and Applications, InTech, Apr. 2011, ISBN: 978-953-307-174-9, doi: 10.5772/14795.

[87] M. Wooldridge, "Introduction to Multi Agent Systems," John Wiley and Sons, ISBN: 978-0470519462.

[88] P. Leitão, J. Mendes, A. Bepperling, D. Cachapa, A.W. Colombo, and F. Restivo, "Integration of virtual and real environments for engineering service-oriented manufacturing systems," Journal of Intelligent Manufacturing, Springer US, Dec. 2012, vol. 23, pp. 2551-2563, doi: 10.1007/s10845-011-0591-8.

[89] U. Inden, N. Mehandjiev, L. Mönch, and P. Vrba, "Towards an Ontology for Small Series Production," Industrial Applications of Holonic and Multi-Agent Systems, Lecture Notes in Computer Science, pp. 128-139 , 2013.

[90] A. Huchzermeier and C. H. Loch, "Project Management Under Risk: Using the Real Options Approach to Evaluate Flexibility in R&D," Management Science, informs, Jan. 2001, vol. 47, Issue 1, pp 85-101, ISSN: 0025-1909, doi: 10.1287/mnsc.47.1.85.10661.

[91] S. Spinler and A. Huchzermeier, "Realoptionen: Eine marktbasierte Bewertungsmethodik für dynamische Investitionsentscheidungen unter Unsicherheit," Controlling und Management, Gabler Verlag, Mar. 2004, vol. 48, Issue 1, pp 66-71, ISSN: 1864-5410, ISBN: 978-3-663-01579-6, doi: 10.1007/BF03255757.

[92] MyBusinessCommunities. *ADVENTURES: Entwicklung von Bewertungsmethoden und Internet-basierten Handelssystemen für Optionen auf Start-Up Ventures, Dienstleistungen und Nicht-lagerfähige Produkte (Finanzdienste)*. [Online]. Available from: http://www.dl2100.de/projectdetail.php?PHPSESSID=e16628...ort1=4&projectid=9/ 2014.11.17

[93] R. Gulati, C. Casto, and C. Krontiris. *How the Other Fukushima Plant Survived*. [Online]. Available from: https://hbr.org/2014/07/how-the-other-fukushima-plant-survived/ 2014.11.17

[94] D. Weinberger, "Too Big To Know: Rethinking Knowledge Now That the Facts Aren't the Facts, Experts Are Everywhere and the Smartest Person in the Room," Basic Books, 2012, ISBN: 978-0-465-02142-0.

[95] E. Brynjolfsson and A. McAfee, "The Second Machine Age," W.W.Norton & Compnay, Jan. 2014, ISBN: 978-0-393-23935-5.

[96] J. Manyika et al.. *Big data: The next frontier for innovation, competition, and productivity*. [Online]. Available from: http://www.mckinsey.com/~/media/McKinsey/dotcom/Insights%20and%20pubs/MGI/Research/Technology%20and%20Innovation/Big%20Data/MGI_big_data_full_report.ashx/ 2014.11.17

[97] C. Anderson. *The End of Theory: The Data Deluge Makes the Scientific Method Obsolete*. [Online]. Available from: http://archive.wired.com/science/discoveries/magazine/16-07/pb_theory/ 2014.11.17

[98] W3C. *SPARQL Query Language for RDF*. [Online]. Available from: http://www.w3.org/TR/rdf-sparql-query/ 2014.11.17

Spatial Trajectory Planning based on Visibility Clustering Analysis in Urban Environments

Oren Gal and Yerach Doytsher
Mapping and Geo-information Engineering
Technion - Israel Institute of Technology
Haifa, Israel
e-mails: {orengal, doytsher}@technion.ac.il

***Abstract*—In this paper, we present a conceptual Spatial Trajectory Planning (STP) method using Rapid Random Trees (RRT) planner, generating visibility motion primitives in urban environments. Visibility motion primitives are set by using Spatial Visibility Clustering (SVC) analysis. Our Spatial Visibility Clustering (SVC) method estimates the number of clusters (groups) k, based on 3D visible volumes analysis in urban environments. Our SVC method proposes fast and exact 3D visible volumes analysis in urban scenes based on an analytic solution. We test and analyze the SVC method by using real records of pedestrians' mobility datasets from the city of Melbourne and by setting control points for efficient monitoring and control using a K-means clustering algorithm.**

Keywords-Visibility; 3D; Spatial analysis; Motion Planning.

I. INTRODUCTION AND RELATED WORK

Spatial clustering in urban environments is a new spatial field from trajectory planning aspects [1]. The motion and trajectory planning fields have been extensively studied over the last two decades [2][4][8][12][14][16][38] [39][40][41][54]. The main effort has focused on finding a collision-free path in static or dynamic environments, i.e., in moving or static obstacles, using roadmap, cell decomposition, and potential field methods [22][50][55].

The path-planning problem becomes an NP-hard one, even for simple cases such as time-optimal trajectories for a system with point-mass dynamics and bounded velocity and acceleration with polyhedral obstacles [9].

Path planning algorithms can be distinguished as local and global planners. The local planner generates one, or a few, steps at every time step, whereas the global planner uses a global search to the goal over a time-spanned tree. Examples of local (reactive) planners are [15][35][47][60]. These planners are too slow, do not guarantee safety and neglect spatial aspects.

Recently, iterative planners [5][16][17][30][48][59] have been developed that compute several steps at a time, subject to the available computation time. The trajectory is generated incrementally by exploring a search-tree and choosing the best branch.

Efficient solutions for an approximated problem were investigated by LaValle and Kuffner, addressing non-holonomic constraints by using the Rapidly Random Trees (RRT) method [39][41]. Over the years, many other semi-randomized methods were proposed, using evolutionary programming [7][43][52].

The randomized sampling algorithms planner, such as RRT, explores the action space stochastically. The RRT algorithm is probabilistically complete, but not asymptotically optimal [32]. The RRT* planner [33] challenges optimality by a rewiring process each time a node is added to the tree. However, in cluttered environments, RRT* may behave poorly since it spends too much time deciding whether to rewire or not.

Overall, only a few works have focused on spatial analysis characters integrated into trajectory planning methods such as visibility analysis or spatial clustering methods [22][55].

'Clustering methods' refers to the division of data sets into groups, each containing similar objects. Data modeling is extensively studied in statistics, mathematics and machine learning [19]. Most of the common clustering methods can be divided into hierarchical and partitioning methods.

Partitioning algorithms determine the clusters directly, such as the well-known K-Means method [27][28], where by a hierarchical mechanism, builds the clusters gradually [24].

Clustering methods of 2D spatial data (such as GIS database) were also studied, defining data proximity by using, inter alia, a Delaunay diagram. These methods focused on performances and low complexity, by keeping K-nearest neighbors using a connectivity graph where clusters become connected components [13][26].

Many clustering methods are based on a significant user's input parameter, the number of clusters k. Over the years, several criteria were introduced to find the optimal k. In the case of k-means clustering method, F-statistic (also known as the F-test) generates the optimal k. Another popular choice of separation measure is a Silhouette coefficient [34].

Our research contributes to the spatial data clustering field, where, as far as we know, visibility analysis has become a leading factor for the first time. The SVC method, while mining the real pedestrians' mobility datasets, enables by a visibility analysis to set the number of clusters.

Analyzing pedestrian's mobility from a spatial point of view mainly focused on route choice [3][29], simulation model [51] and agent-based modeling [25][31][37][57].

The efficient computation of visible surfaces and volumes in 3D environments is not a trivial task. The visibility problem has been extensively studied over the last twenty years, due to the importance of visibility in GIS and Geomatics, computer graphics and computer vision, and robotics. Accurate visibility computation in 3D environments is a very complicated task demanding a high computational effort, which could hardly have been done in a very short time using traditional well-known visibility methods [53].

The exact visibility methods are highly complex, and cannot be used for fast applications due to their long computation time. Previous research in visibility computation has been devoted to open environments using DEM models, representing raster data in 2.5D (Polyhedral model), and do not address, or suggest solutions for, dense built-up areas.

Most of these works have focused on approximate visibility computation, enabling fast results using interpolations of visibility values between points, calculating point visibility with the Line of Sight (LOS) method [10]-[11]. Lately, fast and accurate visibility analysis computation in 3D environments has been presented [18][20][21].

In this paper, we present, for the first time as far as know, a unique conceptual Spatial Trajectory Planning (STP) method based on RRT planner. The generated trajectories are based on visibility motion primitives set by SVC Optimal Control Points (OCP) as part of the planned trajectory, which takes into account exact 3D visible volumes analysis clustering in urban environments.

The proposed planner includes obstacle avoidance capabilities, satisfying dynamics' and kinematics' agent model constraints in 3D environments, guaranteeing probabilistic completeness. The generated trajectories are dynamic ones and are regularly updated during daylight hours due to SVC OCP during daylight hours. STP trajectories can be used for tourism and entertainment applications or for homeland security needs.

The SVC is a unified method for estimating the number of clusters using 3D visible volumes analysis, called Spatial Visibility Clustering (SVC). Based on our previous work, we use a fast and efficient analytic solution, setting visibility boundaries of visible surfaces from the viewpoint. We extend our solution to 3D volumes, computing 3D visible volumes. By using F-criteria, we set the optimal number of clusters from the visibility aspect.

We demonstrate SVC method using real datasets from the city of Melbourne's 24-hours pedestrians monitoring system, localizing control points at each hour during the day, using a K-means algorithm with SVC output, i.e., number of clusters *k*. We analyze pedestrians' mobility behavior and suggest dividing the day into four time zones, based on our datasets and setting optimal control points during these time zones.

In the following sections, we first introduce the RRT planner and our extension for a spatial analysis case, such as 3D visibility. Later, we present the SVC method, and the extended visible volumes analysis and SVC simulation using the city of Melbourne's datasets. We demonstrate the SVC method by dividing daylight hours into four time zones and setting optimal control points. Later on, we present the STP planner, using RRT and SVC capabilities.

II. SPATIAL RAPID RANDOM TREES

In this section, the RRT path planning technique is briefly introduced with spatial extension. RRT was first introduced in [39][41], dealing with high-dimensional spaces by taking into account dynamic and static obstacles including dynamic and non-holonomic robots' constraints.

The main idea is to explore a portion of the space using sampling points in space, by incrementally adding new randomly selected nodes to the current tree's nodes.

RRTs have an (implicit) Voronoi bias that steers them towards yet unexplored regions of the space. However, in case of kinodynamic systems, the imperfection of the underlying metric can compromise such behavior. Typically, the metric relies on the Euclidean distance between points, which does not necessarily reflect the true cost-to-go between states. Finding a good metric is known to be a difficult problem. Simple heuristics can be designed to improve the choice of the tree state to be expanded and to improve the input selection mechanism without redefining a specific metric.

A. RRT Stages

The RRT method [39] is a randomized one, typically growing a tree search from the initial configuration to the goal, exploring the search space. These kinds of algorithms consist of three major steps:

1. **Node Selection:** An existing node on the tree is chosen as a location from which to extend a new branch. Selection of the existing node is based on probabilistic criteria such as metric distance.
2. **Node Expansion:** Local planning applied a generating feasible motion primitive from the current node to the next selected local goal node, which can be defined by a variety of characters.
3. **Evaluation:** The possible new branch is evaluated based on cost function criteria and feasible connectivity to existing branches.

These steps are iteratively repeated, commonly until the planner finds feasible trajectory from start to goal configurations, or other convergence criteria.

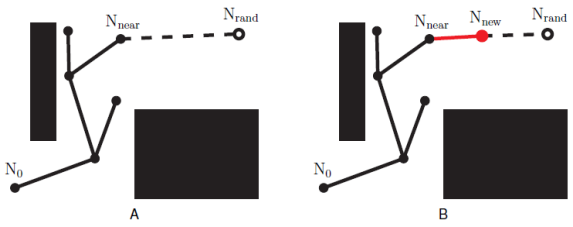


Figure 1. The RRT algorithm: (A) Sampling and node selection steps; (B) Expansion step.

A simple case demonstrating the RRT process is shown in Figure 1. The sampling step selects N_{rand}, and the node selection step chooses the closest node, N_{near}, as shown in Figure 1.A. The expansion step, creating a new branch to a new configuration, N_{new}, is shown in Figure 1.B. An example for growing RRT algorithm is shown in Figure 2.

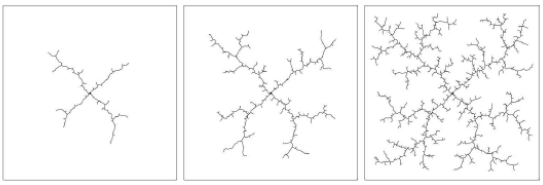


Figure 2. Example for growing RRT algorithm (source [39]).

B. Spatial RRT Formulation

We formulate the RRT planner and revise the basic RRT planner [39] for a 3D spatial analysis case for a continuous path from initial state x_{init} to goal state x_{goal}:

1. **State Space:** A topological space, *X*.
2. **Boundary Values:** $x_{init} \subset X$ and $x_{goal} \subset X$.
3. **Free Space**: A function $D: X \rightarrow \{true, false\}$ that determines whether $x(t) \subset X_{free}$ where X_{free} consist of the attainable states outside the obstacles in a 3D environment.
4. **Inputs:** A set, *U*, contains the complete set of attainable control efforts u_i, that can affect the state.
5. **Incremental Simulator:** Given a current state, $x(t)$, and input over time interval Δt, compute $x(t+\Delta t)$.
6. **3D Spatial Analysis:** A real value function, $f(x; u, OCP_i)$ which specifies the cost to the center of 3D visibility volumes cluster points (OCP) between a pair of points in X .

C. Spatial RRT Formulation

We present a revised RRT pseudo code described in Table I, for spatial case generating trajectory *T*, applying *K* steps from initial state x_{init}. The *f* function defines the dynamic model and kinematic constraints, $\dot{x} = f(x; u, OCPi)$, where *u* is the input and OCP_i set the next new state and the feasibility of following the next spatial visibility clustering point.

TABLE I. SPATIAL RRT PSEUDO CODE

```
Generate Spatial RRT (x_init; K; Δt)
T.init (x_init);
For k = 1 to K do
        x_rand ← random.state();
        x_near ← nearest.neighbor (x_rand; T );
        u ← select.input (x_rand; x_near);
        x_new ← new.state (x_near; u; Δt; f);
        T.add.vertex (x_new);
        T.add.edge (x_near; x_new; u);
End
Return T
```

III. SPATIAL VISIBILITY CLUSTERING (SVC) METHOD

We present, for the first time as far as we know, a unified spatial analysis defining the number of clusters in a data set based on analytic visibility analysis, called Spatial Visibility Clustering (SVC). The output of our method can be efficiently used by common clustering methods (e.g., K-means or hierarchical). The number of clusters in dense environments can be used for civil and security applications in urban environments, based on 3D visibility analysis from points of view.

For the last twenty years, many methods were proposed in order to estimate the number of clusters in data sets [6][23][34][36][46][59]. As previously mentioned by [23], the approaches can be divided into global and local methods.

First, we introduce the main steps of our method and formulate the problem of estimating the number of clusters and the proposed volumes visibility analysis in 3D. Later, we present the analysis of the number of clusters using the SVC method, based on real pedestrians' mobility data sets. Finally, we examine a unique division of a twenty four-hour day into four different time zones in Melbourne [44], for control points based on pedestrians' mobility datasets in a number of points of interest, presented in Figure 3.

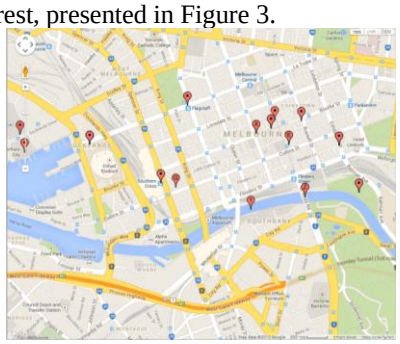


Figure 3. Melbourne Sensors Location for Monitoring Pedestrians' Mobility Data

A. Spatial Visibility Clustering - Main Stages

Our data set $\{X_{ij}\}$, *i=1,2,..,n, j=1,2,..,p,* consists of *p* features measured on n independent viewpoints, marked with blue circles are illustrated in Figure 4. We clustered the data into k clusters, $C_1, C_2, .. C_k$. For cluster r, denote as C_r with n_r viewpoints:

$$V_r = \sum_{i \in C_r} \sum_{j \in C_r} \left\| V(x_i) - V(x_j) \right\| \quad (1)$$

$$V_r = \sum_{i \in C_r} \left\| V(x_i) - V(\bar{x}) \right\|$$

$$T_k = \sum_{r=1}^{k} \frac{1}{S} V_r$$

Where V(x) denotes the visible volumes from a viewpoint *x* bounded inside the total volume *S*, V_r is the sum of the absolute visibility differences of all viewpoints from their cluster visibility mean, and the normalized visible volumes T_k for all clusters *r=1..k*, called ***dispersion***.

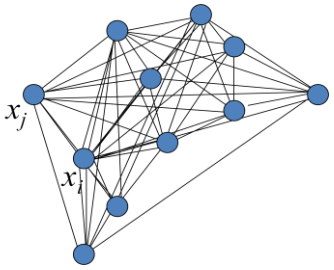


Figure 4. Pedestrians' location architecture based on monitoring datasets, viewpoints marked with blue circles

Similarly to many other methods estimating the number of clusters [23], we define reference data sets distributed uniformly inside bounding volume *S*. We define our reference data sets with the same size of the original data set *X*, and calculate the dispersion of these datasets, T_k^*.

Based on *F* statistic, datasets are analyzed, where adding another cluster does not give a better modeling of the data, also known as *F-test* criteria. By setting a group's visibility variance, the number of clusters can be estimated efficiently:

$$SVC_n(k) = T_k^* - T_k \quad (2)$$

Fast and efficient visibility volume computation from a specific viewpoint, bounded in volume *S*, is presented in the next subsection.

We can summarize SVC steps as follows:

1. Calculate the sum of absolute visibility differences of all points from their cluster visibility mean. Normalize this sum for all possible clusters T_k, also called dispersion.
2. Generate a set of reference datasets, simulated by a uniform distribution model inside bounding volume *S*.
3. Calculate the dispersion of each of these reference datasets, and calculate their mean visibility values.
4. Define SVC for each possible number of clusters as: Expected dispersion of reference datasets - Dispersion of original dataset.

Originally, *F* statistic was used to test the significance of the reduction in the sum of squares as we increase the number of clusters [27]. In general, when the number of clusters increases, the in-cluster decay first declines rapidly. From a certain *k*, dividing a dataset into *k+1* clusters decreases the value of *F-test* function which depends on *k*.

Approximated *F-test* function: Assuming that T_k is the partition of n instances into *k* clusters, and T_{k+1} is obtained from T_k splitting one of the clusters, then the overall mean ratio can be approximated as:

$$F_k = \frac{SVC_n}{SVC_{n+1}} \quad (3)$$

We adapted aspects of previous F statistic theory for visibility analysis. More detailed F statistic analysis can be found in [27].

The spatial meaning of this mathematical clustering formulation can be simplified as a group of viewpoints with minimal difference to the average visible volume in the same bounding box.

B. Analytic 3D Visible Volumes Analysis

In this section, we present fast 3D visible volumes analysis in urban environments, based on an analytic solution which plays a major role in our proposed method of estimating the number of clusters. We extend our previous work [18] for surfaces visibility analysis, and present an efficient solution for visible volumes analysis in 3D.

We analyze each building, computing visible surfaces and defining visible pyramids using analytic computation for visibility boundaries [18][21]. For each object we define Visible Boundary Points and Visible Pyramid.

Visible Boundary Points (VBP) - we define VBP of the object *i* as a set of boundary points *j=1..N_{bound}* of the visible surfaces of the object, from viewpoint *V(x_0, y_0, z_0)*.

$$VBP_{i=1}^{j=1..N_{bound}}(x_0, y_0, z_0) = \begin{bmatrix} x_1, y_1, z_1 \\ x_2, y_2, z_2 \\ .. \\ x_{N_{bound}}, y_{N_{bound}}, z_{N_{bound}} \end{bmatrix} \quad (4)$$

Roof Visibility – The analytic solution for visibility boundaries does not treat the roof visibility of a building [18]. We simply check if viewpoint height *V(z_0)* is lower or higher than the building height $h_{\max_{C_i}}$ and use this to decide if the roof is visible or not:

$$V_{z_0} \geq Z = h_{\max_{C_i}} \quad (5)$$

If the roof is visible, roof surface boundary points are added to VBP. Roof visibility is an integral part of VBP computation for each building.

A simple case demonstrating analytic solution from a visibility point to a building including visible roofs can be seen in Figure 5. The visibility point is marked in black, the

visible parts colored in red, and the invisible parts colored in blue.

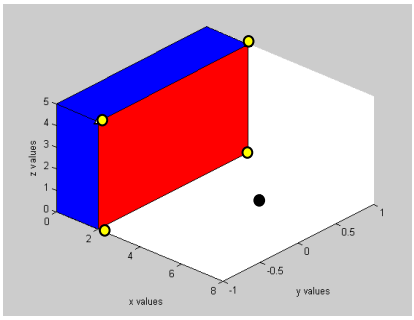


Figure 5. Visibility Volume Computed with the Analytic Solution. Viewpoint is marked in Black, Visible Parts Colored in red, and Invisible Parts Colored in Blue; VBP marked with Yellow Circles

In the previous part, we treated a single building case, without considering hidden surfaces between buildings, i.e., building surfaces (or parts of surfaces) occluded by other buildings, which directly affect the visibility volumes solution. In this section, we introduce our concept for visible volumes inside bounding volume by decreasing visible pyramids and projected pyramids to the bounding volume boundary. First, we define the relevant pyramids and volumes.

The Visible Pyramid (VP): we define $VP_i^{j=1..Nsurf}(x_0, y_0, z_0)$ of the object *i* as a 3D pyramid generated by connecting VBP of specific surface *j* to a viewpoint $V(x_0, y_0, z_0)$.

In the case of a box, the maximum number of N_{surf} for a single object is three. VP boundary, colored with green arrows, can be seen in Figure 6.

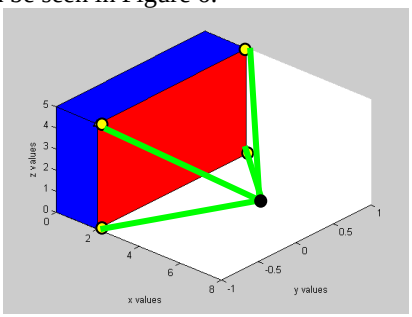


Figure 6. A Visible Pyramid from a Viewpoint (marked as a Black Dot) to VBP of a Specific Surface

For each VP, we calculate Projected Visible Pyramid (PVP), projecting VBP to the boundaries of the bounding volume S.

Projected Visible Pyramid (PVP) - we define $PVP_i^{j..N_{surf}}(x_0, y_0, z_0)$ of the object *i* as 3D projected points to the bounding volume *S*, VBP of specific surface *j* trough viewpoint $V(x_0, y_0, z_0)$. VVP boundary, colored with purple arrows, can be seen in Figure 7.

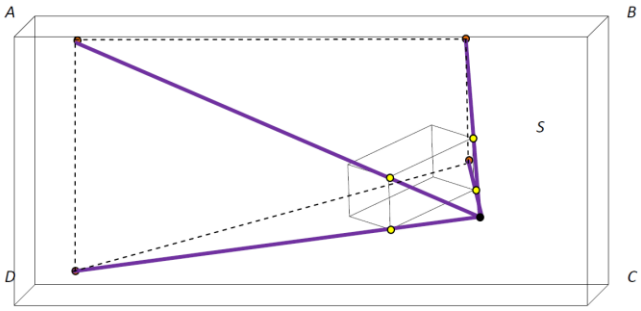


Figure 7. Invisible Projected Visible Pyramid Boundaries colored with purple arrows from a Viewpoint (marked as a Black Dot) to the boundary surface ABCD of Bounding Volume *S*

The 3D Visible Volumes inside bounding volume *S*, VV_S, computed as the total bounding volume *S*, V_S, minus the Invisible Volumes IV_S. In a case of no overlap between buildings, IV_S is computed by decreasing the visible volume from the projected visible volume, $\sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j))$.

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} IV_{S_i}^j \quad (6)$$

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j))$$

By decreasing the invisible volumes from the total bounding volume, only the visible volumes are computed, as seen in Figure 8. Volumes of VPV and VP can be simply computed based on a simple pyramid volume geometric formula.

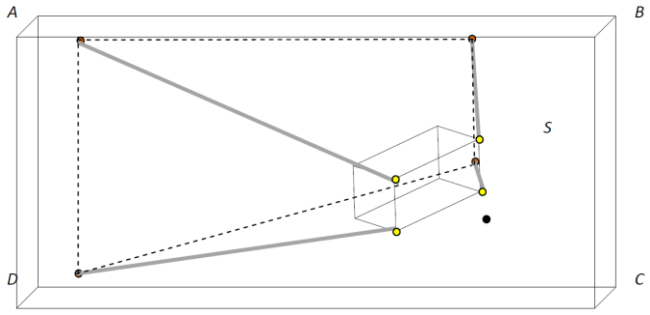


Figure 8. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ Colored in Gray Arrows. Decreasing Projected Visible Pyramid boundary surface ABCD of Bounding Volume S from Visible Pyramid

In a case of two buildings without overlapping, IV_S computed for each building, as presented above, as can be seen in Figure 9.

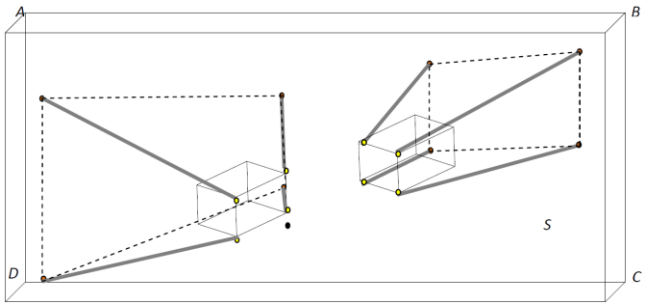


Figure 9. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ Colored in Gray Arrows. Decreasing Projected Visible Pyramid boundary surface ABCD of Bounding Volume S from Visible Pyramid

Considering two buildings with overlap between object's Visible Pyramids, as seen in Figure 10(a). In Figure 10(b), VP_1^1 boundary is colored by green lines, VP_2^1 boundary is colored by purple lines and the hidden and Invisible Surface between visible pyramids $IS_{VP_1^i}^{VP_2^i}$ is colored in white.

Invisible Hidden Volume (IHV) - We define Invisible Hidden Volume (*IHV*), as the *Invisible Surface (IS)* between visible pyramids projected to bounding box *S*.

For example, IHV in Figure 10(c) is the projection of the invisible surface between visible pyramids colored in white, projected to the boundary plane of bounding box *S*.

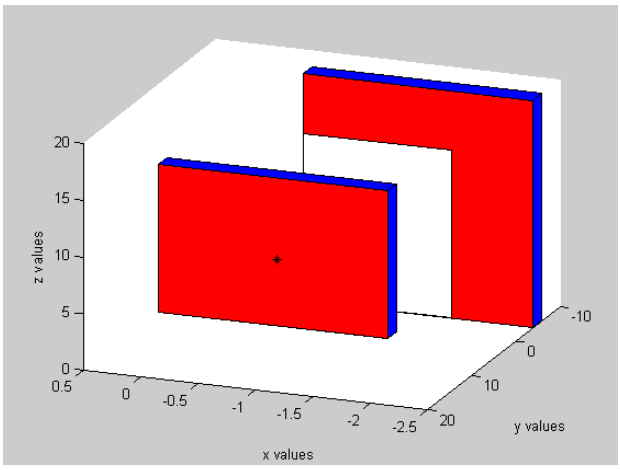
In the case of overlapping buildings, by computing invisible volumes IV_S, we decrease IHV twice between the overlapped objects, as can be seen in Figure 10(c), *IHV* boundary points denoted as $\{A_{11},..,A_{18}\}$. The same scene is presented in Figure 11, where Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ is colored in purple and green arrows for each building.

The *PVP* of the object close to the viewpoint is marked in black, colored with pink circles denoted as boundary set points $\{B_{11},..,B_{18}\}$ and the far object's *PVP* is colored with orange circles, denoted as boundary set points $\{C_{11},..,C_{18}\}$. It can be seen that *IHV* is included in each of these invisible volumes, where $\{A_{11},..,A_{18}\} \in \{B_{11},..,B_{18}\}$ and $\{A_{11},..,A_{18}\} \in \{C_{11},..,C_{18}\}$.

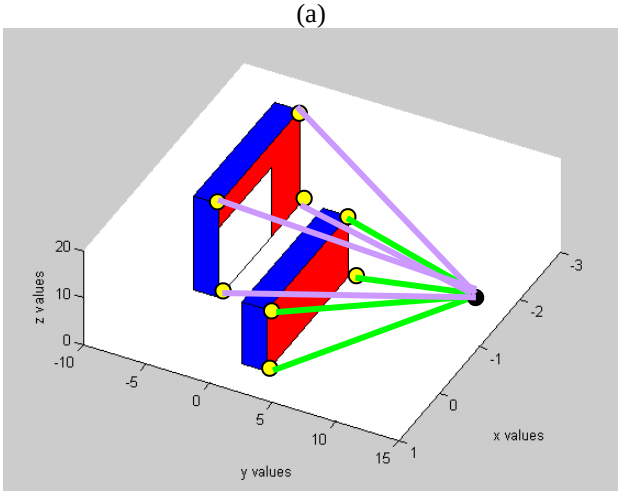
Therefore, we add *IHV* between each overlapping pair of objects to the total visible volume. In the case of overlapping between objects' visible pyramids, 3D visible volume is formulated as:

$$VV_S = V_S - \sum_{i=1}^{N_{obj}} \sum_{j=1}^{N_{surf}} (V(PVP_i^j) - V(VP_i^j) + IHV_i^j) \qquad (7)$$

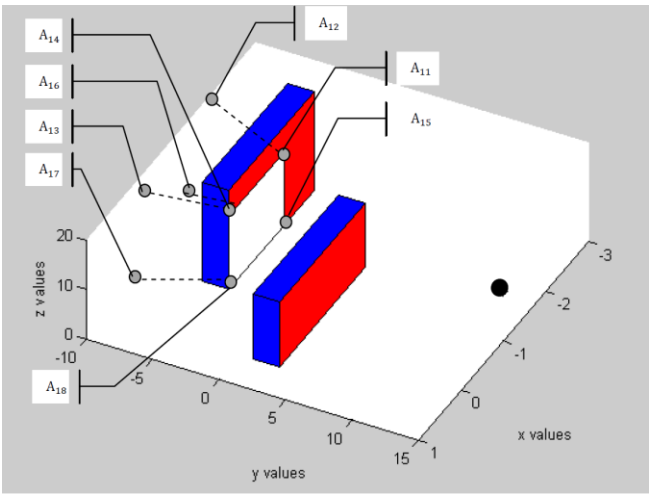
The same analysis holds true for multiple overlapping objects, adding the IHV between each two consecutive objects.



(a)



(b)



(c)

Figure 10. (a) Computing Hidden Surfaces between Buildings , VP_2^1 Base Plane, $IS_{VP_1^i}^{VP_2^i}$ (b) The Two Buildings - VP_1^1 in green and VP_2^1 in Purple (from the Viewpoint) and $IS_{VP_1^i}^{VP_2^i}$ in White (c) IHV boundary points colored with gray circles denoted

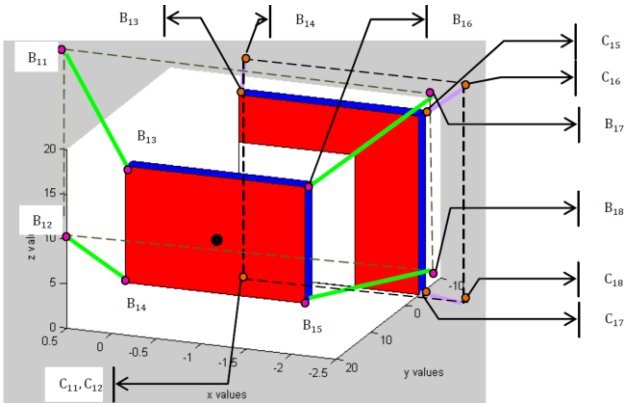


Figure 11. Invisible Volume $V(PVP_i^j) - V(VP_i^j)$ colored in purple and green arrows for each building. PVP of the object close to viewpoint colored in black, colored with pink circles and the far object PVP colored with orange circle

In Figure 12, we demonstrate the case of three buildings with overlapping. The invisible surfaces are bounded with dotted lines, while the projected visible surfaces to the overlapped building are colored in gray. In order to calculate the visible volumes from a viewpoint, *IHV* between each two buildings must be added as a visible volume, since it is already omitted at the previous step as an invisible volume.

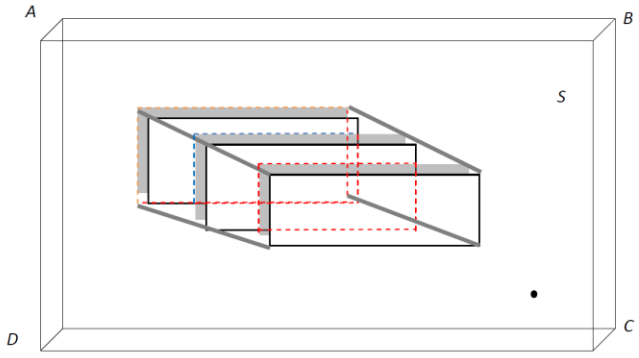


Figure 12. Three overlapping buildings. Invisible surfaces bounded with dotted lines, projected visible surfaces of the overlap building colored in gray

C. Simulations

In this section, we demonstrate the SVC method of estimating the number of clusters based on pedestrians' mobility datasets. For each pedestrian's location datasets, we analyze the 3D visible volumes inside bounding volume *S*, defined as a 3D box.

Our datasets are based on the city of Melbourne's 24-hour pedestrian monitoring system (24PM). This system measures pedestrian activity at several Points of Interests (POI) with counting sensors. Pedestrian mobility datasets are available online with interactive maps, as seen in Figure 13, and can be downloaded for a specific date.

Figure 13. City of Melbourne's 24-hour pedestrian monitoring system (24PM) – Online Visualization Map

Our datasets include the number of pedestrians in each hour during the 2nd of July 2013, at different seventeen points of interest in Melbourne where counting sensors are located and defined as viewpoints. Based on these datasets, we approximated the pedestrians' location using the well-known and common kinematic model for pedestrians presented by Hoogendoorn et al. [29]. Based on this model, pedestrian 2D location can be estimated as:

$$x(t + \Delta t) = x(t) + V(t)\Delta t + w \quad (8)$$

where *w* is a white noise of a standard Wiener Process which reflects the uncertainty in the expected traffic condition, described as Gaussian distribution.

Pedestrian speed *V* can be divided into three major groups:

1. Fast: 1.8 meters per second
2. Standard: 1.3 meters per second
3. Slow: 0.8 meters per second

$$V(t) \sim N(\mu = 1.3, \sigma^2 = 0.5)$$
$$w \sim \sqrt{\Delta t} N(0,1) \quad (9)$$

The kinematic model of a pedestrian is only a part of the estimation and prediction of his movement in an urban environment. For simplicity, we use only a kinematic model for a pedestrian's future location, since decision-making in this field is very complicated.

At time step t, pedestrian location $x(t)$, is taken from a specific POI from our dataset, and the estimated pedestrian location $x(t + \Delta t)$ can be computed. In our simulations we set Δt for five minutes. For example, pedestrians' 2D location in UTM coordination, using the Hoogendoorn etc. model [29], between 6-7 a.m., can be seen in Figure 14.

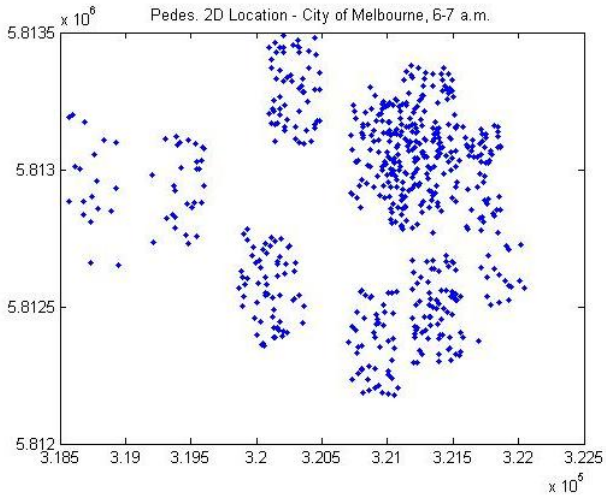


Figure 14. Pedestrians' 2D estimated location using the Hoogendoorn etc. model [29] between 6-7 a.m.

Each of pedestrian locations is processed as a viewpoint for estimating the number of clusters from spatial visibility aspects. The 3D visible volumes computation presented in the previous section are applied for computing T_k, as described in Section III.

At each POI, we set the reference dataset of the pedestrian location distributed uniformly around the POI location, where the reference dataset size is the same one as the original dataset for the same POI, computing T_k^*.

We set the possible number of clusters from one to ten, demonstrating the SVC method. The number of clusters based on visible volumes analysis per day hour is presented in Figure 15.

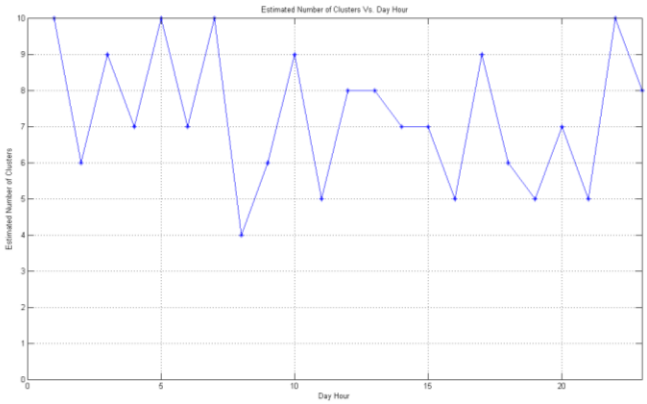


Figure 15. Number of Clusters for each Hour of 2/7/2013 Using SVC

As we can see in Figure 15, there is a correlation between the number of clusters and the pedestrians' mobility behavior. The number of clusters is close to the maximum (ten clusters in our case) during 6-9 AM, as can be predicted due to pedestrians' mobility while going to work. The number of clusters drops to a figure between eight to four clusters during the midday hours, and climbs again during nigh hours. More incentives analyzing pedestrians' mobility patters are presented in the next section.

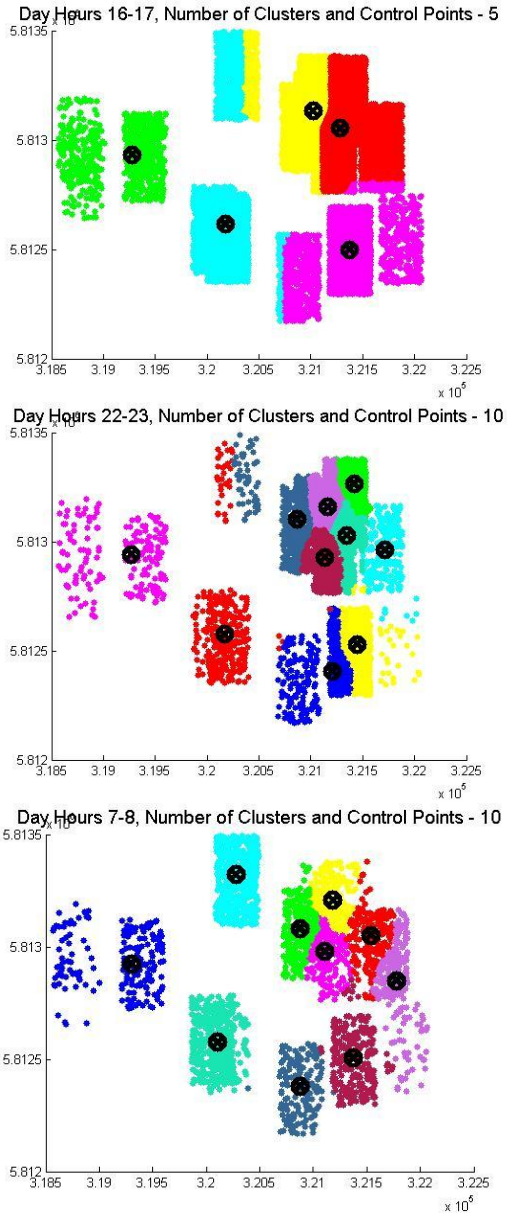


Figure 16. Control Points Location and Clusters Presentation during Each Hour in a Day. Control points are marked with black circles. Pedestrians' mobility Clustered in different colors

IV. ANALYZING PEDESTRIANS' MOBILITY DATASETS

A. Control Points

In this section, we analyze pedestrians' mobility datasets during one day, estimating the number of clusters by using the SVC outcome, which is based on visibility analysis. Upon that, we use the K-means clustering method.

K-means clustering intends to partition n objects into k clusters, where each object belongs to the cluster with the nearest mean. The centroid of all objects in each cluster

is set as control point. This method produces exactly k different clusters, where k is predefined from the SVC method. The objective of K-means clustering is to minimize total intra-cluster variance, or the squared error function. K-means algorithm stages can be described as:

1. Cluster the data into *k* groups, where *k* is predefined from the SVC method.
2. Select *k* points at random as cluster centers.
3. Assign objects to their closest cluster center using Euclidean distance function.
4. Calculate the centroid all objects in each cluster.
5. Repeat steps 2, 3 and 4 until the same points are assigned to each cluster.

By using K-means and SVC method, control points location can be seen in Figure 16.

It can be noticed in Figure 16 that in some cases the geometric location of the sensor location is separated into two different clusters. Our maximal number of clusters is set to ten, whereas there are seventeen sensors. We set the maximal number of clusters to be smaller than the number of sensors on the scene. One of the major contributions of our work, related to the adaptive clustering capability, is separating datasets into a different clustering and setting the control points from a visibility aspect. Moreover, control point location should cover more than one area, as can be seen in Figure 14, and also depends on pedestrians' mobility during this hour, as can be seen in the next sub-section.

Video simulations showing control points locations using K-means clustering and SVC methods are available in [56].

B. *Time Zones*

In this section, we concentrate on learning pedestrians' patterns for setting *Optimal Control Points (OCP)*, i.e., control points for each time zone. We divide the day into four time zones for efficient pedestrian monitoring:

1. Morning hours (movement to work) – 6 – 9 AM.
2. Mid-Day Hours (between morning and afternoon) – 10 AM to 16 PM.
3. Afternoon hours (back from work and activity hours) – 17- 20 PM.
4. Night hours 20 – 23 PM.

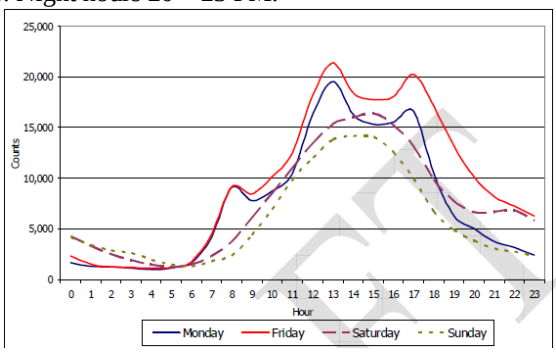


Figure 17. Pedestrian Activity Analysis [45]

The suggested division of time zones partition can also be seen clearly in an official pedestrian monitoring report of the city of Melbourne [45] (see Figure 17). The number of pedestrians counted by the monitoring system rises at the suggested time zones.

In order to get reliable and comprehensive results regarding pedestrian mobility patterns, we tested a full month's (July 2013) dataset, analyzing each day for twenty-four hours.

Based on the average estimated number of clusters using SVC on these datasets, we found out that the number of optimal control points during these time zones is:

- Morning hours – Nine control points
- Mid-Day Hours – Six control points
- Afternoon hours – Seven control points
- Night hours – Eight control points

The localization of the optimal control points and the number of clusters for each time zone can be seen in Figure 18.

It can be seen that in the different time zones, three optimal control points and their cluster division are almost identically marked with arrows and numbers in Figure 18.

Four optimal control points with similar clustering can be seen in three time zones in Figure 18. These results can also be applicable for personal-security and homeland security application in urban environments, localizing forces and sensors for optimal monitoring and trajectory planning during a daylight hours.

V. SPATIAL TRAJECTORY PLANNING (STP)

In this section, we present a conceptual STP method based on RRT planner. The method generates visibility motion primitives in urban environments. The STP method is based on a RRT planner extending the stochastic search to specific *OCP*. These primitives connecting between nodes through OCP are defined as visibility primitives.

A common RRT planner is based on greedy approximation to a minimum spanning tree, without considering either path lengths from the initial state or following or getting close to specific OCP. Our STP planner consist of a tree's extension for the next time step with probability to goal and probability to waypoint, where trajectories can be set to follow adjacent points or through OCP. The planner includes obstacle avoidance capabilities, satisfying dynamics' and kinematics' agent model constraints in 3D environments. As we demonstrated in the previous section, the OCP are dynamic during daylight hours. Due to *OCP's* dynamic character, the generated trajectory is also a dynamic one during daylight hours.

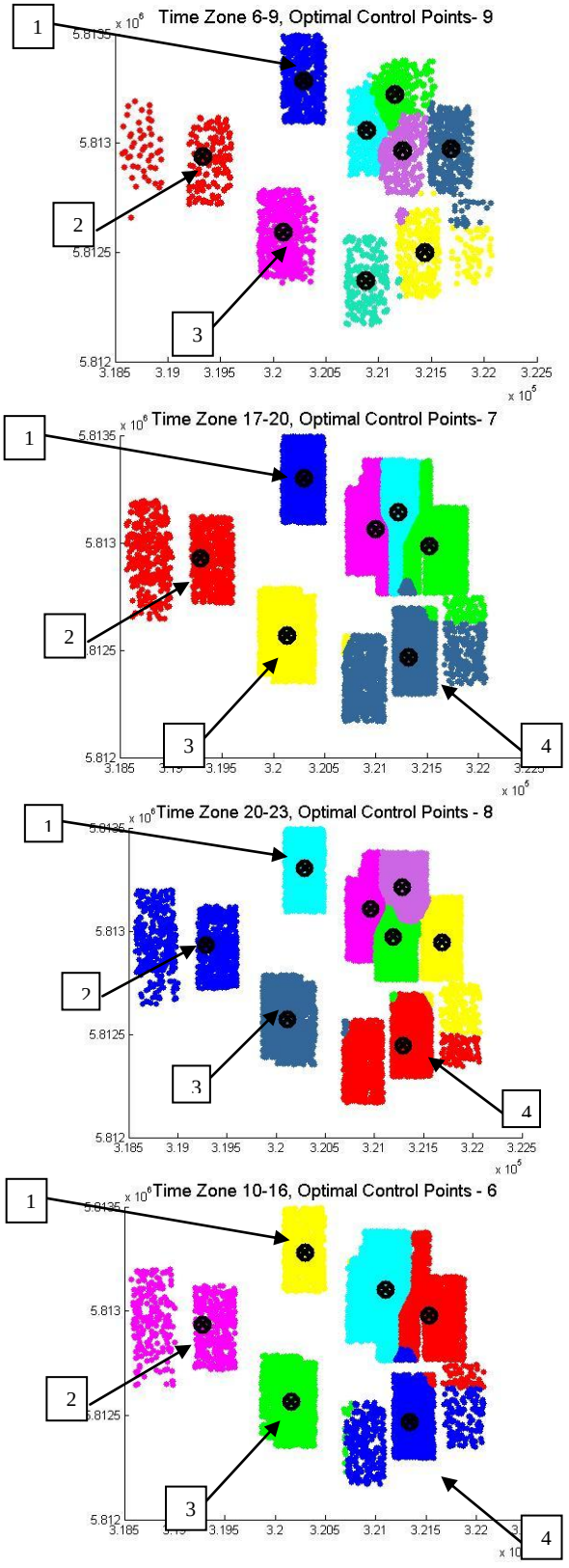


Figure 18. Optimal Control Points Location in Four Time Zones. Optimal Control points marked with black circles. Pedestrians' mobility Clustered in different colors

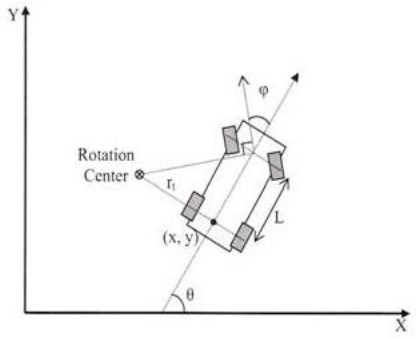


Figure 19. Four-Wheeled Car Model with Front-Wheel Steering [Lewis]

We present our concept addressing the STP method formulating planner for a UGV model, integrating *OCP's* as part of the generated trajectories along with obstacle avoidance capability.

A. Dynamic Model

In this section, we suggest an Unmanned Ground Vehicle (UGV) dynamic model based on the four-wheeled car system (UGV) with rear-wheel drive and front-wheel steering [42]. This model assumes that only the front wheels are capable of turning and the back wheels must roll without slipping, and all the wheels turn around the same point (rotation center) which is co-linear with the rear axle of the car, as can be seen in Figure 19, where *L* is the length of the car between the front and rear axles. r_t is the instantaneous turning radius.

Thus, UGV dynamic model can be described as:

$$\dot{x} = f(x,u) = \begin{Bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{Bmatrix} = \begin{Bmatrix} v\cos(\theta) \\ v\sin(\theta) \\ \frac{v}{L}\tan(\emptyset) \end{Bmatrix} \quad (10)$$

The state vector, *x*, is composed of two position variables (x,y) and an orientation variable, θ. The x-y position of the car is measured at the center point of the rear axle. The control vector, *u*, consists of the vehicle's velocity, *v*, and the angle of the front wheels, ϕ, with respect to the car's heading.

B. Search Method

Our search is guided by following spatial clustering points based on 3D visible volumes analysis in 3D urban environments, i.e., Optimal Control. The cost function for each next possible node (as the target node) consists of probability to closest *OCP*, P_{OCPi} , and probability to random point, P_{rand} .

In case of overlap between a selected node and obstacle in the environment, the selected node is discarded, and a new node is selected based on P_{OCPi} and P_{rand}. Setting the probabilities as P_{OCPi} =0.9 and P_{rand}=0.1, yield to the exploration behavior presented in Figure 20.

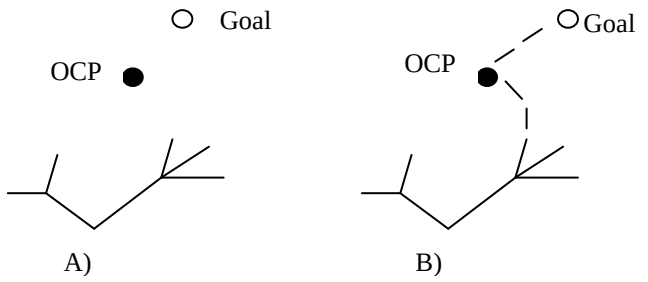


Figure 20. STP Search Method: (A) Start and Goal Points; (B) Explored Space to the Goal Through OCP

C. STP Planner Pseudo-Code

We present our STP planner pseudo code described in Table II, for spatial case generating trajectory T with search space method presented in the Section V.B. The search space is based on P_{OCPi} and P_{rand}. We apply K steps from initial state x_{init}. The f function defines the dynamic model and kinematic constraints, $\dot{x} = f(x; u)$, where u is the input and $OCPi$ are local target points between start to goal states.

TABLE II. STP PLANNER PSEUDO CODE

```
STP Planner (x_init; x_Goal ;K; Δt; OCP)
T.init (x_init);
x_rand ← random.state();
x_near ← nearest.neighbor(x_rand; T );
u ← select.input(x_rand; x_near);
x_new ← new.state.OCP (OCP_1; u; Δt; f);
While x_new ≠ x_Goal do
        x_rand ← random.state();
        x_near ← nearest.neighbor(x_rand; T );
        u ← select.input(x_rand; x_near);
        x_new ← new.state.OCP (OCP_i; u; Δt; f);
        T.add.vertex(x_new);
        T.add.edge(x_near; x_new; u);
end
return T;

Function new.state.OCP (OCP_i;u; Δt; f)
Set P_OCPi , Set P_rand
p←uniform_rand[0..1]
if   0 < p < P_OCPi
      return x_new = f(OCP_i,u,Δt);
else
     if P_OCPi < p < P_rand+ P_OCPi
then
      return RandomState();
end.
```

D. Completeness

Motion-planning and search algorithms commonly describe 'complete planner' as an algorithm that always provides a path planning from start to goal in bounded time. For random sampling algorithms, 'probabilistic complete planner' is defined as: if a solution exists, the planner will eventually find it by using random sampling. In the same manner, the deterministic sampling method (for example, grid-based search) defines completeness as resolution completeness.

Sampling-based planners, such as the STP planner, do not explicitly construct search space and the space's boundaries, but exploit tests with preventing collision with obstacles and, in our case, taking spatial considerations into account. Similarly, to other common RRT planners, which share similar properties with the STP planner, our planner can be classified as a probabilistic complete one.

VI. CONCLUSIONS

In this paper, we have presented a unique planner concept, STP, generating trajectory in 3D urban environments based on UGV model. The planner takes into account obstacle avoidance capabilities and passes through optimal control points calculated from spatial analysis. The spatial analysis defines the number of clusters in a dataset based on an analytic visibility analysis, named SVC.

The SVC method is based on fast and efficient 3D visible volumes computation. Estimating the number of clusters is based on minimum normalized visible volumes to reference datasets distributed uniformly inside bounding volume S. We demonstrated the SVC by using datasets from the city of Melbourne's 24-hour pedestrian monitoring system (24PM).

In the second part of this research, based on the SVC-estimated number of clusters, we analyzed pedestrians' mobility behavior, setting control points during daylight hours and dividing a daylight hours into four time zones. We found a correlation of several optimal control points in different time zones.

Based on similar spatial analysis in other urban scenes, one can set optimal control points for various applications, such as entertainment events that can be efficiently visible at such points, or monitoring crowds' movements from these control points in emergencies, planning medical assistance.

The STP concept includes probabilistically complete properties which changes dynamically during daylight hours. The planner allows us to generate trajectory for various applications such as personal security and homeland security applications in urban environments, localizing police forces and sensors for optimal monitoring at different hours of a day.

Future work will focus on simulation in real data records using the STP planner, generating trajectories in 2D and 3D urban environments using an Unmanned Aerial Vehicle (UAV) model. Future research will also include performances and algorithm complexity analysis for STP and SVC methods.

VII. REFERENCES

[1] O. Gal and Y. Doytsher, "Spatial Visibility Clustering Analysis In Urban Environments Based on Pedestrians' Mobility Datasets," The Sixth International Conference on Advanced Geographic Information Systems, Applications, and Services, pp. 38-44, 2014.

[2] J. Bellingham, A. Richards, and J. How, "Receding Horizon Control of Autonomous Aerial Vehicles," in Proceedings ofthe IEEE American Control Conference, Anchorage, AK, pp. 3741–3746, 2002.

[3] A. Borgers and H. Timmermans, "A model of pedestrian route choice and demand for retail facilities within inner-city shopping areas," Geographical Analysis, vol. 18, No. 2, pp. 115-128, 1996.

[4] S. A. Bortoff, "Path planning for UAVs," In Proc. of the American Control Conference, Chicago, IL, pp. 364–368, 2000.

[5] O. Brock and O. Khatib, "Real time replanning in high-dimensional configuration spaces using sets of homotopic paths," In Proc. of the IEEE International Conference on Robotics and Automation, San Francisco, CA, pp. 550-555, 2000.

[6] R. B. Calinski and J. Harabasz, "A Dendrite Method for Cluster Analysis," Communications in Statistics, vol. 3, pp. 1–27, 1974.

[7] B. J. Capozzi and J. Vagners, "Navigating Annoying Environments Through Evolution," Proceedings of the 40th IEEE Conference on Decision and Control, University of Washington, Orlando, FL, 2001.

[8] H. Chitsaz and S. M. LaValle, "Time-optimal paths for a Dubins airplane," in Proc. IEEE Conf. Decision. and Control., USA, pp. 2379–2384, 2007.

[9] B. Donald, P. Xavier, J. Canny, and J. Reif, "Kinodynamic Motion Planning," Journal of the Association for Computing Machinery, pp. 1048–1066, 1993.

[10] Y. Doytsher and B. Shmutter, "Digital Elevation Model of Dead Ground," Symposium on Mapping and Geographic Information Systems (Commission IV of the International Society for Photogrammetry and Remote Sensing), Athens, Georgia, USA, 1994.

[11] F. Durand, "3D Visibility: Analytical Study and Applications," PhD thesis, Universite Joseph Fourier, Grenoble, France, 1999.

[12] M. Erdmann and T. Lozano-Perez, "On multiple moving objects," Algorithmica, Vol. 2, pp. 477–521, 1987.

[13] V. Estivill-Castro and I. Lee, "AMOEBA: Hierarchical Clustering Based on Spatial Proximity Using Delaunay Diagram," In Proceedings of the 9th International Symposium on Spatial Data Handling, Beijing, China, 2000.

[14] P. Fiorini and Z. Shiller, "Motion planning in dynamic environments using velocity obstacles," Int. J. Robot. Res. vol. 17, pp. 760–772, 1998.

[15] W. Fox, D. Burgard, and S. Thrun, "The dynamic window approach to collision avoidance," IEEE Robotics and Automation Magazine, vol. 4, pp. 23–33, 1997.

[16] T. Fraichard, "Trajectory planning in a dynamic workspace: A 'state-time space' approach," Advanced Robotics, vol. 13, pp. 75–94, 1999.

[17] E. Frazzoli, M.A. Daleh, and E. Feron, "Real time motion planning for agile autonomous vehicles," AIAA Journal of Guidance Control and Dynamics, vol. 25, pp. 116–129, 2002.

[18] O. Gal and Y. Doytsher, "Fast and Accurate Visibility Computation in a 3D Urban Environment," in Proc. of the Fourth International Conference on Advanced Geographic Information Systems, Applications, and Services, Valencia, Spain, pp. 105-110, 2012 [accessed February 2014].

[19] P. Arabie and L. J. Hubert, "An Overview of Combinatorial Data Analysis," in Arabie, P., Hubert, L.J., and Soete, G.D. (Eds.) Clustering and Classification, pp. 5-63, 1996.

[20] O. Gal and Y. Doytsher, "Fast Visibility Analysis in 3D Procedural Modeling Environments," in Proc. of the, 3rd International Conference on Computing for Geospatial Research and Applications, Washington DC, USA, 2012.

[21] O. Gal and Y. Doytsher, "Fast Visibility in 3D Mass Modeling Environments and Approximated Visibility Analysis Concept Using Point Clouds Data," Int. Journal of Advanced Computer Science, IJASci, vol 3, vo 4, April 2013, ISSN 2251-6379, [accessed February 2014].

[22] O. Gal and Y. Doytsher, "Fast and Efficient Visible Trajectories Planning for Dubins UAV model in 3D Built-up Environments," Robotica, FirstView, Article pp. 1-21 Cambridge University Press 2013 DOI: http://dx.doi.org/10.1017/S0263574713000787, [accessed February 2014].

[23] A. Gordon, Classification (2nd ed.), London: Chapman and Hall/CRC Press, 1999.

[24] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," In Proceedings of the ACM SIGMOD Conference, Seattle, WA, pp. 73-84, 1998.

[25] M. Haklay, D. O'Sullivan, and M.T. Goodwin, "So go down town: simulating pedestrian movement in town centres," Environment and Planning B: Planning & Design, vol. 28, no. 3, pp. 343-359, 2001.

[26] D. Harel and Y. Koren, "Clustering spatial data using random walks," In Proceedings of the 7th ACM SIGKDD, San Francisco, CA, pp. 281-286, 2001.

[27] J. Hartigan, "Clustering Algorithms". John Wiley & Sons, New York, NY, 1975.

[28] J. Hartigan and M. Wong, "Algorithm AS136: A k-means clustering algorithm," Applied Statistics, vol. 28, pp. 100-108, 1979.

[29] S. P. Hoogendoorn and P. H. L. Bovy, "Microscopic pedestrian way finding and dynamics modelling," In Schreckenberg, M., Sharma, S.D. (eds.) Pedestrian and Evacuation Dynamics. Springer Verlag: Berlin, pp. 123-154, 2001.

[30] D. Hsu, R. Kindel, J-C. Latombe, and S. Rock, "Randomized kinodynamic motion planning with moving obstacles," Algorithmics and Computational Robotics, vol. 4, pp. 247–264, 2000.

[31] B. Jiang, "SimPed: Simulating pedestrian flows in a virtual urban environment," Journal of Geographic Information and Decision Analysis, vol. 3, no. 1, pp. 21-30, 1999.

[32] S. Karaman and E. Frazzoli, "Sampling-based algorithms for optimal motion planning," Int. J. Robot. Res., vol. 30, no. 7, pp. 846–894, 2011.

[33] S. Karaman, M. Walter, A. Perez, E. Frazzoli, and S. Teller, "Anytime motion planning using the RRT*," in Proc. IEEE Int. Conf. Robot. Autom., Shanghai, pp. 1478–1483, May 2011.

[34] L. Kaufman and P. Rousseeuw, "Finding Groups in Data: An Introduction to Cluster Analysis," John Wiley and Sons, New York, NY, 1990.

[35] N.Y. Ko and R. Simmons, "The lane-curvature method for local obstacle avoidance," In International Conference on Intelligence Robots and Systems, 1998.

[36] W. J. Krzanowski and Y. T. Lai, "A Criterion for Determining the Number of Groups in a Data Set Using Sum of Squares Clustering," Biometrics, vol. 44, pp. 23–34, 1985.

[37] M.P. Kwan, "Analysis of human spatial behavior in a GIS environment: recent developments and future prospects," Journal of Geographical System, no. 2, pp. 85-90, 2000.

[38] J. C, Latombe, "Robot Motion Planning,", Kluwer Academic Press, 1990.

[39] S. M. LaValle, "Rapidly-exploring random trees: A new tool for path planning," TR 98-11, Computer Science Dept., Iowa State University, 1998.

[40] S. M. LaValle, "Planning Algorithms," Cambridge,U.K.:Cambridge Univ. Press, 2006.

[41] S. M. LaValle and J. Kuffner. "Randomized kinodynamic planning," In Proc. IEEE Int. Conf. on Robotics and Automation, Detroit, MI, pp. 473–479, 1999.

[42] L.R. Lewis, "Rapid Motion Planning and Autonomous Obstacle Avoidance for Unmanned Vehicles," Master's Thesis, Naval Postgraduate School, Monterey, CA, December 2006.

[43] C. W. Lum, R. T. Rysdyk, and A. Pongpunwattana, "Occupancy Based Map Searching Using Heterogeneous Teams of Autonomous Vehicles," Proceedings of the 2006 Guidance, Navigation, and Control Conference, Autonomous Flight Systems Laboratory, Keystone, CO, August 2006.

[44] Melbourne City: http://www.pedestrian.melbourne.vic.gov.au/#date=31-08-2013&time=9 [accessed February 2014].

[45] Melbourne Report: https://docs.google.com/file/d/0B380gpj_lbU-dnRaRTFXdlh5Znc/edit [accessed February 2014].

[46] G.W. Milligan and M. C. Cooper, "An Examination of Procedures for Determining the Number of Clusters in a Data set," Psychometrika, vol. 50, pp. 159–179, 1985.

[47] J. Minguez and L. Montano, "Nearest diagram navigation. a new realtime collision avoidance approach," In International Conference on Intelligence Robots and Systems, 2000.

[48] J. Minguez, N. Montano, L. Simeon, and R. Alami, "Global nearest diagram navigation," In Proc. of the IEEE International Conference on Robotics and Automation, 2002.

[49] B. Moulin, W. Chaker, and J. Perron, "MAGS project: Multi-Agent GeoSimulation and Crowd Simulation," Kuhn, W., Worboys, M.F. and Timpf, S. (Eds.): LNCS 2825, pp. 151–168, 2003.

[50] K. J. Obermeyer, "Path Planning for a UAV Performing Reconnaissance of Static Ground Targets in Terrain," in Proceedings of the AIAA Guidance, Navigation, and Control Conference, Chicago, 2009.

[51] S. Okazaki and S. Matsushita, "A study of simulation model for pedestrian movement with evacuation and queuing," Proceedings of the International Conference on Engineering for Crowd Safety, London, UK, pp. 17-18, March 1993.

[52] A.Pongpunwattana and R.T. Rysdyk, "Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments," AIAA Journal of Aerospace Computing, Information, and Communication, pp. 580–604, 2004.

[53] H. Plantinga and R. Dyer, "Visibility, Occlusion, and Aspect Graph," The International Journal of Computer Vision, vol. 5, pp. 137-160, 1990.

[54] J. Sasiadek and I. Duleba, "3d local trajectory planner for uav," Journal of Intelligent and Robotic Systems, vol. 29, pp. 191–210, 2000.

[55] V. Shaferman and T. Shima, "Co-evolution genetic algorithm for UAV distributed tracking in urban environments," in ASME Conference on Engi- neering Systems Design and Analysis, July 2008.

[56] https://sites.google.com/site/orenusv/home/svc [accessed February 2014].

[57] T. Schelhorn, D. Sullivan, and M. Haklay, "STREETS: An agent-based pedestrian model," http://www.casa.ucl.ac.uk/streets.pdf., 1999, [accessed February 2014].

[58] C. Stachniss and W. Burgard, "An integrated approach to goal directed obstacles avoidance under dynamic constrains for dynamic environment," In International Conference on Intelligence Robots and Systems, 2002.

[59] R. Tibshirani, G. Walther, and T. Hastie, "Estimating the Number of Clusters in a Dataset via the Gap Statistic," Journal of the Royal Statistical Society, Ser. B, vol. 32, pp. 411–423, 2001.

[60] L. Ulrich and J. Borenstien, "Vfh+: Reliable obstacle avoidance for fast mobile robots," In Proc. of the IEEE International Conference on Robotics and Automation, 1998.

Combining Association Mining with Topic Modeling to Discover More File Relationships

Namita Dave, Karen Potts, Vu Dinh, and Hazeline U. Asuncion
School of Science, Technology, Engineering, and Mathematics
University of Washington Bothell
Bothell, WA, USA
{namitad , pottsk2 , vdinh143, hazeline}@u.washington.edu

Abstract— **Software maintenance tasks require familiarity with the entire software system to make proper changes. Often, maintenance engineers who did not develop the software are tasked with corrective or adaptive maintenance tasks. As a result, modifying the software becomes a time-consuming process due to their lack of familiarity with the source code. To help software engineers locate relevant files for a maintenance task, association mining has been used to identify the files that frequently change together in a software repository. However, association mining techniques are limited to the amount of project history stored in a software repository. We address this difficulty by using a technique that combines association mining with topic modeling, referred to as Frequent Pattern Growth with Latent Dirichlet Allocation (FP-LDA). Topic modeling aims to uncover file relationships by learning semantic topics from source files. We validated our technique via experiments on seven open source projects with different project characteristics. Our results indicate that FP-LDA can find more related files than association mining alone. We also offer lessons learned from our investigation.**

Keywords-Association mining; Topic Modeling; Software Engineering.

I. INTRODUCTION

Software maintenance has been known to incur the highest cost among the different phases in the software lifecycle [1, 2]. This may be due to an engineer's unfamiliarity with the software to modify, requiring more time to understand the source code [3]. Maintenance tasks also become more difficult as the complexity of the code increases and as code degradation occurs over time, due to patches and workarounds [4].

To assist with software maintenance tasks, various techniques have been proposed to find related source code, including static and dynamic analyses, recommendation systems, and code search techniques. Static analysis techniques, more specifically, dependency analysis, provide file relationships based on call graphs [5]. Dynamic analysis tools, meanwhile, are able to identify relationships between files based on execution traces [6]. These techniques, however, are generally language-specific. Recommendation systems, meanwhile, provide possible files of interest based on a developer's past activities, textual similarity, check-in records, or email records [7, 8]. These systems generally use information retrieval techniques, along with user context, to provide files of interest. Code search techniques find related code based on syntactic or structural matches [9].

Association mining is another technique used to find related files. Association mining uncovers relationships between files, based on files that have been modified together in the past. This technique generates rules, which specify which files are frequently changed together. Unlike the other techniques, association mining is not specific to the programming language used or restricted to syntactic or structural matches of a query.

The most commonly used algorithms for association mining are Apriori [10] and Frequent Pattern Growth (FP-Growth) [11]. While these algorithms provide some level of accuracy, they are highly dependent on the project history. If there are not enough modifications in the software project, or if the modifications are sparse throughout the software system, there are fewer chances that association mining will result in correct rules.

Meanwhile, machine learning techniques, such as Latent Dirichlet Allocation (LDA), allow us to automatically detect relationships between files based on semantic similarity. LDA is an unsupervised statistical approach for learning semantic topics from a set of documents [12]. It is a fully automated approach that does not require training labels. It only requires a set of documents and number of topics to learn.

Thus, we aim to address the challenges of association mining by combining it with LDA. Our technique, Frequent Pattern Growth with Latent Dirichlet Allocation (FP-LDA), allows us to achieve better recall results than solely using association mining. By combining these two techniques, we are able to overcome the limitations of each technique. LDA allows us to find file associations even with limited modification history. Association mining, meanwhile, allows us to find associations among files where semantic similarities may not be readily apparent. We previously introduced FP-LDA [1] but this paper provides more details regarding our technique.

The contributions of this research paper are as follows: (1) combination of association mining and topic modeling to identify file relationships in a software project, (2) experiments on seven open source projects, and (3) lessons learned in effectively using these techniques. We also created a set of tools that automates the entire process—from pre-processing the data to querying related files.

We envision our technique being used in the following scenarios: a developer presented with a modification task knows at least one file to change and would like to know what other files to change; a technical lead wishes to ensure that changes performed by a teammate are complete; a maintenance engineer needs to perform impact analysis to determine the feasibility of changing a section of the code.

The rest of the paper is organized as follows. Section II covers background on association mining and Section III covers background on topic modeling. In Section IV, we present our combined approach, FP-LDA. We then validate our approach in Section V. Section VI covers lessons learned. We conclude with future work.

II. ASSOCIATION MINING

This section covers background on association mining, selection of the association mining technique, application, limitations, and related work.

A. Background

Association rule mining is a method used to discover patterns in large data sets. Initially, it was used in Market Basket Analysis to find how items bought by customers are related [13]. Rules are mined from the dataset, such as "Customers who bought item A also bought item B". In the case of mining file associations in software projects, rules such as "Developers who modified file A also modified file B" are mined [14]. In order to mine these rules, patterns must be analyzed in the dataset.

We now discuss the main ideas of association mining based on work by Agrawal and Srikant [10], as applied to software development.

$$I = \{i_1, i_2, ..., i_m\} \quad (1)$$

Let (1) represent the total set of items. In this paper, the files in the repository are items. T represents a set of transactions

$$T = \{t_1, t_2, ..., t_n\}, \quad (2)$$

which are in the software repository being mined. Each transaction t is a set of items such that $t \subseteq I$. In this paper, t represents one atomic commit.

Given the set of transactions T (see (2)), the goal of association mining is to find all the association rules that have support and confidence greater than the user specified threshold values. An itemset is a collection of items. The support is defined as the fraction of transactions that contain the itemset and from which the rule is derived. The confidence denotes the strength of a rule. An association rule is represented as

$$X \rightarrow Y \text{ [support = 20\%, confidence = 80\%]} \quad (3)$$

In this notation, itemset X is called the antecedent and itemset Y is called the consequent such that $X, Y \subseteq I$. Both antecedent and consequent are comprised of one or more items. Assume that both X and Y consist of one file, each namely x and y, respectively. Then, this rule says that in 20% of the check-in transactions, both x and y files are modified and the transactions, which changed file x, also changed file y 80% of the time.

The threshold support value specified by the user is called minimum support. This is an important element that makes association mining practical. It reduces the search space by limiting the number of rules generated [15]. The threshold confidence value specified by the user is called minimum confidence [15].

There are two types of measures for association mining: objective and subjective. Support and confidence, which we just discussed, are objective measures of association mining [16]. Subjective measures are unexpectedness and actionability [17]. The generated rules are "unexpected" or surprising if the relationship is not obvious to the user. For example a file customers.h is, most of the time, going to change if customers.c is modified. Such a rule, though valid has little usefulness to the developer. However, if the recommendations help the developer to perform her task effectively, then such rules have high actionability. Actionability refers to the capability of the approach to yield a rule that can be acted upon with some advantage.

B. Selection of Association Mining Technique

There are two commonly used association mining techniques. The first sequential pattern mining algorithm used to mine rules was Apriori algorithm [10]. Later, the Frequent Pattern Growth Algorithm, or FP-Growth, was introduced [11]. We now discuss the ideas behind these two techniques and the rationale for selecting FP-Growth.

Apriori is a classic algorithm for learning association rules over transactional databases for sample collections of items bought by customers [10]. It works in two steps. In the first step, it generates the candidate itemsets. These are the set of items that have the minimum support. In the second step, association rules are generated. Apriori uses the property that any subsets of a frequent itemset are also frequent. The essential idea behind Apriori algorithm is that it iteratively generates candidate itemsets of length (k + 1) from frequent itemsets of length k and then tests their corresponding frequency in the database. Apriori is not efficient when used with large data sets, as generation of candidate item sets and support counting is very expensive, as confirmed in [18].

FP-Growth is a faster and more scalable approach to mine a complete set of frequent patterns by pattern fragment growth. This can be achieved by using a compact prefix tree structure for storing a transaction dataset [11]. This algorithm operates in two steps. In the first step, it creates a compact Frequent Pattern tree to encode the database. The construction of an FP-tree begins with pre-processing the input data with an initial scan of the database to count support for single items. The single items that do not meet the threshold support values are eliminated. The database is then scanned for the second time to produce an initial FP-

tree. The second step runs a depth first recursive procedure to mine the FP-tree for frequent itemsets with increasing cardinality. The FP tree stores a single item at each node. The root node of an FP tree is empty. The path from the root to a node in the FP tree is a subset of the transactions database. The items in the path are in decreasing order of support. In the second step, the algorithm examines a conditional-pattern base for each itemset starting with length 1 and then constructs its own conditional FP-tree. Unlike the Apriori algorithm, it avoids generating expensive candidate itemsets. Each conditional FP-tree is recursively mined to generate frequent itemsets. The algorithm uses a divide-and-conquer approach to decompose the mining task into smaller tasks of mining the confined conditional databases. Interested readers can refer to work by Han, Pei and Yin for more information [11].

C. Application

Association mining has been used in the past to support various software engineering tasks. Some approaches to accomplish these tasks rely on structural analysis of code while others rely on textual mining. In mining associations from software projects, two data sources are primarily used as sequence-sources: the project history and the code structure. In cases where history or documentation is unavailable, the structure of the software may be analyzed by breaking it into groups, further decomposing them into entities, and then mining association rules from the entity sets [19]. MAPO, a tool for suggesting API usage patterns, analyzes sequences in code structure found within open-source repositories [20]. Clustering techniques have also been explored to find similarities in program entities in order to support software maintenance [21]. Techniques that rely on the structure of the software are useful in cases where one language is used or when the interoperation of processes is not a concern. In analyzing open source repositories, we have found that several types of code may be checked-in together. In addition, interoperating processes may not share dependencies in source descriptions, and yet they may pass messages, and thus rely upon each other.

Another source of data for mining associations between source files is found in the history logs of configuration management systems, such as those found in the open source repositories that we have mined. Association mining with FP-Growth has been applied to these change histories [14]. We build on this approach and we enhance this technique with the use of topic modeling. An example of a tool that performs association mining on history logs is Rose [19]. It is a tool that parses syntactic entities from the committed source code, such as classes, functions, and fields. Association mining is applied to this parsed version of the history data set. The association rules obtained could predict that programmers who changed a given entity also changed the recommended entities. Our approach is similar, in that we mine rules from the history of the repository. However, we do not provide the fine granularity of connection provided by parsing syntactic entities, for the reasons outlined above, relating to techniques that rely on software structure to mine associations. Instead, we use topic modeling techniques to find associations between source files based on variable names, comments, and other information available as plain text. By restricting the words we use in our topic model, this technique is applicable to any source code language. Later, we discuss our approach to pre-processing source code and our approach to language-specific keywords (see Section IV.A). Association rules have also been mined from repository histories in order to find traceability links [22]. As pointed out by David et al., mining the project history has the benefit of reducing the need to rely on the content of the data in instances where it may be sparse or where the content of related artifacts is not related. While we do not rely on the content, we do leverage it where appropriate with topic modeling. It has been pointed out that temporal information can also be useful in eliminating false-positive recommendations [23]. However, this was not applicable to our approach, since we consider both the history of aggregated commits as well as the content of the source files.

D. Limitations

Association mining is useful in finding patterns in the data that satisfy minimum support and minimum confidence constraints. However, some researchers have shown that association mining often results in redundant and unimportant rules. A drawback is that it is difficult to eliminate insignificant rules [24].

In this research, the number of association rules generated depends on the amount of modification history of a project. Also, there is a possibility that not all modules or files may be changed during a software maintenance phase. This can affect the number of rules generated.

E. Related Work

Our work is most closely related to previous work in mining frequently changed files from a software repository [14, 25]. We used association mining as other software engineering researchers have used this technique in the past. We build on top of this existing work and examine the benefits of combining association mining with topic modeling. While others have used collaborative filtering [26], we use topic modeling, which is a probabilistic version of matrix factorization over the word-document matrix. In this paper, we use topic modeling to analyze the semantic content of source code and commit comments. In previous work, we have used topic modeling to identify associations between various software files and architecture components [27]. In the future, we plan to use topic modeling to identify associations between files and authors. Our work is also related to other techniques that seek to identify relationships between software files, such as recommendation systems, code search techniques, and dependency analysis.

Recommendation systems for software engineering may also recommend files for modification. Not all recommendation systems use association rule mining, but eRose a plugin for Eclipse does [8]. The common factor among all recommendation systems for software engineering is that they rely on the user's context in order to provide recommendations. While recommendation systems may help

find related files in source code, the issue of user context is outside of the scope of our work.

Code search techniques may also be used to find source files that are related to one another. These techniques have their roots in traditional information retrieval methods [28]. An equivalency study was undertaken to compare various IR methods in the area of traceability recovery [29]. The results of this study showed that while Latent Semantic Indexing (LSI), Jensen-Shannon (JS), and Vector Space Model (VSM) provided higher accuracy in identifying related files, LDA was able to capture associations, which the other methods could not. Recent work in code search has been performed to enhance the accuracy of these methods by allowing the user to specify both the syntactic and semantic properties of a search [28]. Code search techniques, however, fall short in finding relationships between project files, which are not semantically or syntactically related. Meanwhile, our technique finds these relationships based on the change history of the project and semantic relationship.

Dependency analysis tools may be used to find relationships between source files based on call graphs [5]. By making use of the project histories, we can mine relationships between any files that are checked-in together, as opposed to simply analyzing the code structure. As discussed previously, we also have the ability to find relationships between source code files written in different languages. Most importantly, this approach helps to detect cross cutting concerns in which there may be a relationship between two files, but no relationship in a call-graph. For example, a project created for multiple operating systems may contain two source files, which accomplish the same task, but have no relationship in the calling tree. In this case, dependency analysis cannot detect these relationships, but our approach can, because of the semantic similarity between files.

III. TOPIC MODELING

This section covers background on topic modeling, how we selected the topic modeling technique, application, and limitations.

A. Background

LDA is an unsupervised statistical approach for learning semantic topics from a set of documents [12]. Since it is an unsupervised machine learning technique, no training labels are necessary. This is a fully automated approach that only requires a set of documents and the number of topics to learn

LDA is a generative Bayesian topic model for a corpus of documents. The basic concept behind LDA is that it discovers topics. Then, it associates a set of words with each topic. Lastly, it defines each document as a probabilistic mixture of these topics. Thus, each document can belong to multiple topics. Additional details regarding LDA's generative process are in [12].

Here are some concepts used in LDA:

- A word is a basic unit of discrete data.
- A document is characterized by a vector of word counts.
- A corpus has a total of W words in its vocabulary.
- D documents placed side by side, gives W x D matrix of counts.
- A topic is a probability distribution over W words.
- Each document is associated with a probability distribution over T topics.

To obtain a semantic interpretation of a topic, we simply examine the highest-probability words in that topic. For example, if a topic has high probability words "window", "dialog", "height", "width", "button", we can infer the topic to be related to the user interface of the software.

As we discuss in the next section, we use LDA to determine possible relationships between source code files through their topic distributions. Each source code file equates to a document in LDA.

B. Selection of Topic Modeling Technique

Topic modeling algorithms generally fall under two categories: sampling-based and variational methods [12]. Sampling-based algorithms collect samples to approximate the posterior with an empirical distribution. Variational methods, meanwhile, use a parameterized family of distributions and then find the member of the family that is closest to the posterior. In this paper, we use a fast version of Collapsed Variational Inference (CVB0) for LDA [30], which has been shown to be among the fastest and most accurate methods for learning topic models.

C. Application

Topic modeling has generally been used to analyze unstructured text [31]. In software engineering, topic modeling has been used to relate code topics to authors [32], to enhance the prospective capture of traceability links [27], to derive coupling metrics between classes [33], to find duplicate bug reports [34], and to analyze code fragmentation on the Android framework [35, 36]. Other studies have also shown that LDA can capture associations between software files that are not captured by other IR techniques [37, 29]. In this paper, we use topic modeling to obtain additional file associations to those that can be acquired from mining a project history.

D. Limitations

LDA has generally been applied to unstructured text [31]. Meanwhile, source code is a highly structured text that has a limited range of semantic concepts. The results are also subject to parameters used in LDA. As a result, researchers have examined ways to fine-tune the parameters [36].

We processed the source code prior to running LDA such that reserved words are removed and only semantically meaningful words are used. Our pre-processing technique is similar to the pre-processing technique described here [38].

IV. COMBINED APPROACH

Our technique, FP-LDA, aims to lower the dependency of the result on the project history and to provide an alternative means of uncovering related files. FP-LDA consists of the following steps: (1) data extraction and pre-processing, (2) association data mining, (3) topic modeling, and (4) result querying. Figure 1 shows a high level process

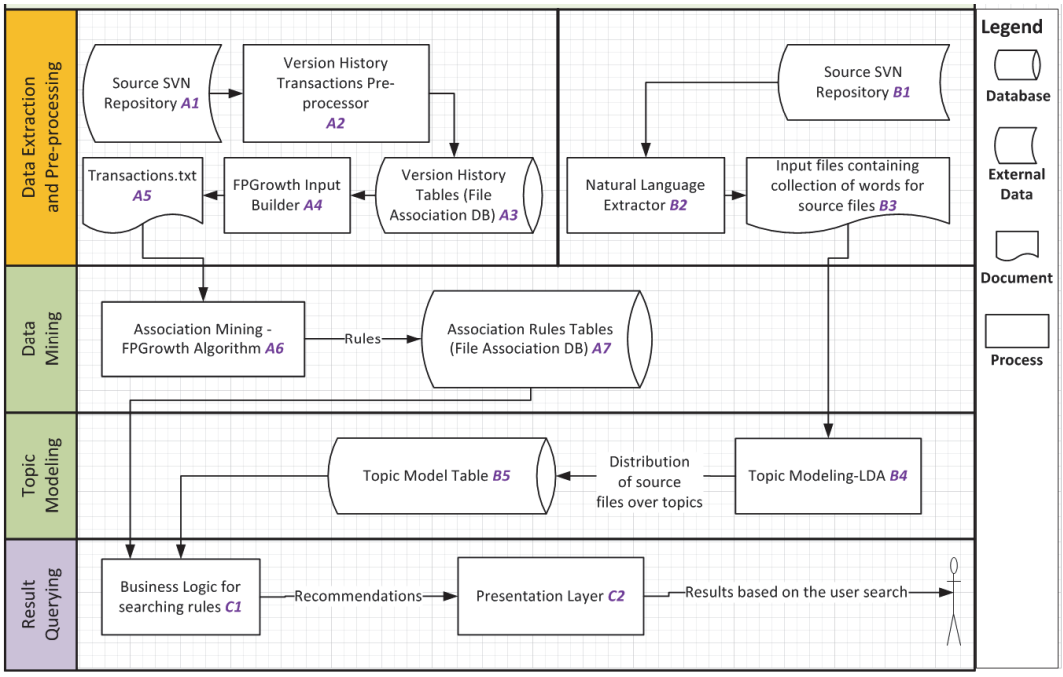


Figure 1. FP-LDA data flow to find file dependencies.

of our technique. Each layer in Figure 1 corresponds to each of these steps. All the processes represented by a rectangle have been implemented.

A. Data Extraction and Pre-processing

Pre-processing version history for data mining. This first step involves extracting the version history of an open source project and preparing the data to be fed as input to the mining algorithm (Step A2). We created a tool that accesses the version history of the project, processes it, and stores the history data in MySQL database. For projects using Subversion (SVN), we used SVNKit application programming interfaces (APIs) [39] to access the version history of the project. SVNKit is an open source Java-based SVN library. For projects using Git, we used JavaGit [40] API to access the version history.

Data pre-processing is an important step in that it removes all unwanted data that may impact data mining (A2). In our technique, our goal is to create a generic pre-processing step to support different open source projects. Thus, we used the following conditions when determining the type of transactions to include in our association mining. Similar to [14], we do not include transactions with more than one hundred files since these transactions may contribute to noise. Such commits may be due to specialized tasks, such as formatting all source code files and then checking-in all files together. We also removed transactions that do not assist in identifying relationships between files, such as single file commits, non-source code commits (e.g., graphic files), and commits of deleted files. The remaining valid transactions are then stored in a database (A3). This is the dataset that will be analyzed by the mining algorithm. We then transform this dataset into a file format that conforms to expected format of the mining algorithm (A4).

Pre-processing source files for LDA. While the mining algorithm examines the entire commit history, we use topic modeling to extract topics from the latest version of the source code. We extracted from the source code semantically meaningful text, such as comments, identifier names and string literals. These words provide clues on the purpose or functionality of the code (B2).

To extract these words, we run each file through a tokenizer. The tokenizer aids in splitting words with underscore or in camel case to obtain the name of objects or variables. We also specified a set of stop words that are programing language-reserved words, and high frequency terms in a software project (see Lessons Learned in Section VI for a detailed discussion). We also removed words like "get" and "set" since source files contain methods that start with these words. This requires some knowledge of the programming language syntax. Another option is to generate the Abstract Syntax Tree using tools like ANTLR [41] to support multiple languages. The generated tree can then be explored to extract the comments and identifiers inside the source code.

B. Association Data Mining

Once the data is preprocessed, we run the data mining algorithm (A6). We used Frequent Pattern Growth (FP-Growth) algorithm for association mining, more specifically, the Liverpool University Computer Science – Knowledge Discovery in Data (LUCS-KDD) implementation of FP-Growth. This Java implementation uses tree structures for association mining [42]. This algorithm requires an input file for the transactions to be analyzed. Each line in the input

Table I. Open Source Project Characteristics.

Project	Repos	Total LOC	Java LOC	No of Files	No of Commits	Years of History
ArchStudio5	Git	838675	194766	2406	596	3.8
ArgoUML	SVN	432521	328494	2254	14922	9.3
EclipseFP	Git	453362	106798	1435	2419	9
Eu_Geclipse	Git	503152	333389	3196	3356	7.5
Lucene	SVN	3893658	1058795	15463	11374	4.5
Thrift	Git	332158	27468	1561	3723	6.3
Xerces	SVN	1266177	260242	2497	5434	14.8

file constitutes one transaction. Each item in the line denotes a file ID. This implementation only works on numeric data. The input file was generated in the previous step. The minimum support and minimum confidence values can be passed as input parameters to the algorithm. The algorithm then generates all the association rules.

In our approach, we did not restrict the frequent itemsets generated to two so that we could uncover more complex relationships. In this case, rules are produced with more than one item in the antecedent and consequent. A file can be related to different files in a different way. If we input only one file, it will give all the recommendations that many not be valid for a certain transaction. However, if the user knows more than one file to be modified for a task, we can refine the predictions. For example, using complex rules we can find out which files change given that two input files are modified together. We store the generated frequent itemsets in a database (A7).

C. Topic Modeling

Once the source files are pre-processed, we extract semantic topics using LDA (B4). We used the CVB0 implementation of LDA [30]. Our implementation of LDA has the following parameters: number of topics and number of iterations. Number of topics is the number of topics we specify. The greater the number of topics, the more fine-grained will be the generated topics. Number of iterations is the number of times the algorithm will run. The higher number of iterations increases the likelihood that the topics will converge. We observed that it is sufficient to run the topic model using 1000 iterations.

D. Result Querying

The last step is to query the results of both the rules generated from association mining and the document relationship to topics (C1). We assume that the user is aware of at least one file that has to be modified for a given modification task. This file is used as the input. The output will show all the files that are recommended or predicted to change along with the input file.

V. VALIDATION

In this section, we discuss how we assess our technique. We cover the setup of our experiment,

A. Experiment with Open Source Projects

In order to validate the ability of FP-LDA to identify relevant files to modify, we conducted experiments on open source projects. We compared FP-LDA with our baseline, FP-Growth.

1) Experiment Setup

We conducted an experiment on seven open source projects that use SVN or Git repositories (see Table I). We selected these projects because these are active projects with different lengths of time (ranging from 3.8 years to almost 15 years) and different range of files (ranging from one thousand files to more than fifteen thousand files).

For each project, we used the same set of parameters. For association mining, we used minimum support of 10 and 15 and confidence value of 40. For topic modeling, we used 25, 50, and 100 topics. For all topic model runs, we used 1000 iterations. We also used topic cutoff values of 10%, 25%, 50%, and 75%. The LDA recommendations were calculated by returning files that have a topic distribution percentage higher than the cutoff for a given topic. We assumed that a file with a higher distribution is semantically closer to a given topic. The validation was performed for these four cutoff percentages. For this experiment, we used Java source code for the topic model.

2) Procedure

To measure the effectiveness of our approach, we used precision and recall. Precision measures the conciseness of a recommendations provided by the approach. Recall measures how many relevant recommendations are made by using this approach. We followed the same approach as used by Ying et al [14]. In this case study, we have assumed that developer is aware of at least one file for a given modification task. Therefore, we specified only one file f_s for generating recommendations for a modification task m. As explained

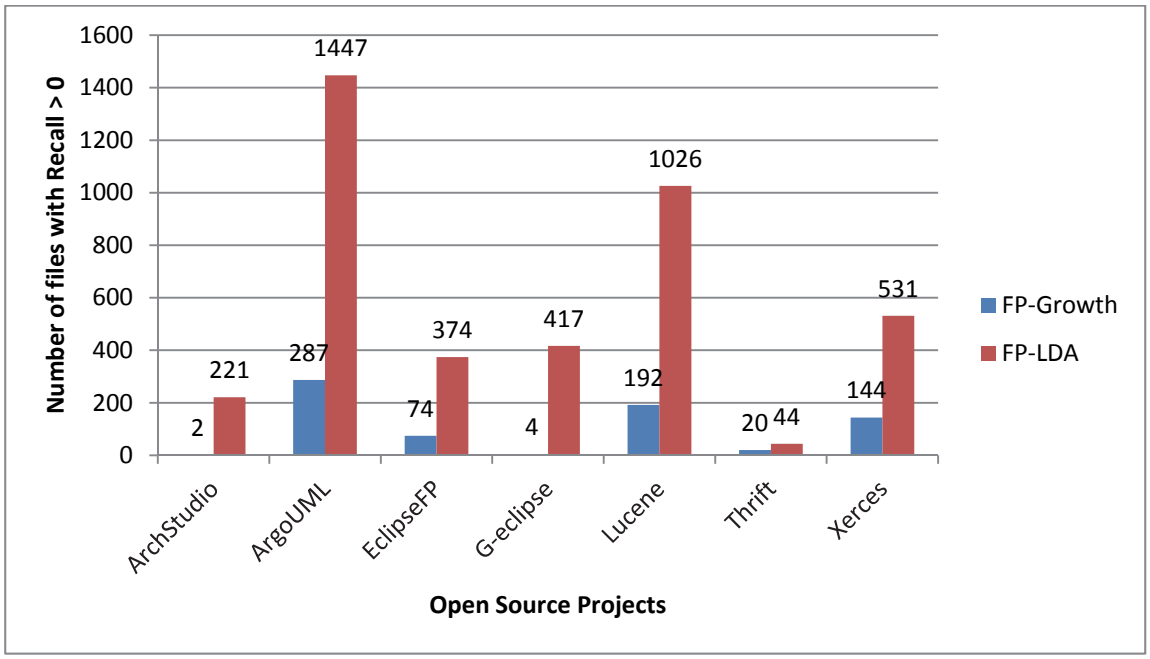


Figure 2. Comparison of recall count between FP-Growth and FP-LDA for the different open source projects.

in [14], the precision $precision(m, f_s)$ of a recommendation $recom(f_s)$ is the fraction of files that are predicted correctly and are part of the solution $f_{sol}(m)$ for the modification task m. The recall $recall(m, f_s)$ of a recommendation $recom(f_s)$ is the fraction of files recommended out of $f_{sol}(m)$.

For example, let us consider a modification task that requires changing files {a, b, c, d}. In addition, let us assume that the recommendations obtained for file b using our approach are files {a, c}. In this case, the precision for file b in this modification task is 100% as the approach recommended correct files. The recall value for file b for same modification task is 66.67% because the approach could predict only two files {a, c} out of {a, c, d}.

In order to determine the effectiveness of our prediction algorithm, we generated FP-Growth rules using 90% of the commit transactions. We then calculated the precision and recall rates of the generated rules on the remaining 10% of the commit transactions (the withheld set). We split the dataset based on time, since this simulates actual practice. Then, we calculated precision and recall for both FP-Growth and FP-LDA for each file in each transaction. We calculated precision and recall for the different parameter combinations. Finally, we counted the number of files in the 10% commit that was able to predict at least one relevant file across the different parameters settings.

3) Experiment Results

As Figure 2 shows, FP-LDA consistently improves FP-growth's ability to identify related files. This is best illustrated in the case of the ArchStudio project where only four rules were produced by FP-Growth using minimum support of 10 and confidence of 40. FP-Growth did not produce any rules with minimum support of 15 and confidence of 40. The small number of rules is due to the fact that ArchStudio is a young project that does not have sufficient history to determine file associations. In this case, FP-LDA improves the ability to identify relevant files by two orders of magnitude. The Thrift project, which shows the least number of improvements of 100%, is due to the fact that this project contains the least number of Java source files in comparison to the other projects, only 1/10th of the project's source code. FP-LDA does not only benefit new projects, but older projects like Xerces, which has almost 15 years of history. In this project, we see an almost 300% increase in its ability to find relevant files.

B. Experiment on Two Open Source Projects

In order to validate the ability of FP-LDA to rank relevant files to modify, we conducted experiments on two open source projects: ArgoUML and EclipseFP. We compared FP-LDA with two baselines, FP-Growth and LDA.

1) Experiment Setup

We selected ArgoUML and EclipseFP projects for implementing ranking.

For ranking the association rules, we used confidence of the rule as a measure to return the recommended files. The higher the confidence, the higher are the chances that predicted files co-occur with the input file in the commit transactions. We selected the top 5 recommendations from FP-Growth. Based on our experience with these projects, the number of association rules generated is not many.

Therefore, by selecting top 5 predictions we covered almost all the predictions that can be provided by FP.

Then for LDA, we used cosine similarity measure to assess the similarity between the source files. The higher value of cosine between two files, the stronger correlation exists between the files. To compare the files using this method we represented each file with total number of frequent words. This is calculated by multiplying the distribution percentage with total number of words in a preprocessed file. We selected the top 10 recommendations from LDA. Next section discusses cosine similarity implementation in detail.

2) Procedure

Though the distribution percentage gives an indication of semantic closeness of a file and topic, the number of recommended files was very high. Therefore, for LDA, we decided to rank the recommended files and return only a limited number of recommendations.

Cosine similarity is often used in text mining applications to assess the similarity between two files [43]. Mathematically, cosine similarity is a measure of how similarity between two vectors and is measured by the cosine of the angle between them.

The cosine similarity between two vectors $\vec{v}$ and $\vec{w}$ of dimension N is calculated

$$\cos(\vec{v},\vec{w}) = \frac{\vec{v}.\vec{w}}{|\vec{v}||\vec{w}|} = \frac{\sum_{i=1}^{N} v_i \times w_i}{\sqrt{\sum_{i=1}^{N} {v^2}_i}\sqrt{\sum_{i=1}^{N} {w^2}_i}} \quad (4)$$

The vectors are derived by multiplying the distribution percentage of a file for a topic with total number of words in a preprocessed file. This gives us a total number of frequent words from a topic present in the file. We assume two files are most similar if the cosine value between them is highest. To obtain topic model recommendations, for a given input file, we recommend files in order of highest cosine similarity. We limit the number of recommendations to 15 so that the precision calculated with FP-LDA is not less than the one obtained using just FP-Growth.

In this setup, we calculated precision and recall for the transactions where FP-Growth resulted in at least one prediction. We also calculated the number of files for all the transactions in test dataset that has at least one correct recommendation.

3) Results

Figure 3 shows the average precision obtained using FP-Growth, FP-LDA and LDA for ArgoUML and EclipseFP. We return at most 15 recommendations in this setup. A low value of precision for FP-LDA may be a result of the fact that most transactions do not consist of 15 files. However, we can see from Figure 4 that the total number of files for the transaction dataset that have at least one correct prediction is greater with FP-LDA than with FP-Growth alone. This means there are more recommendations obtained by using FP-LDA.

C. Examples

In addition to comparing the number of files in the withheld set, which resulted in at least one relevant recommendation, we also examined specific transactions to compare FP-LDA with our baseline. We selected one example that best illustrates each case.

Case 1: FP-Growth recall > 0 and FP-LDA recall > FP-Growth recall: In the Lucene project, we see an example of this in revision ID 1580463 (see Table II). In this example, FP-Growth was run with minimum support of 10 and a minimum confidence of 40%. The topic distribution cutoff was kept at 75%. Figure 5 shows the files committed as part of this transaction. We used Overseer.java as an input file for the techniques.

The association mining alone is able to predict two out of these eight files resulting in a recall value of 0.29. Figure 5 shows the predictions obtained using FP-Growth.

With combined FP-LDA, there are four correct recommendations resulting in a higher recall value of value of 0.6. The precision (0.2) of FP-LDA is low because there were 24 files predicted. The ranking approach we used in Section V.B allows recommendations to be returned that are semantically closer to the topic first and limits the total recommendations. Figure 5 also shows the recommendations from using FP-LDA.

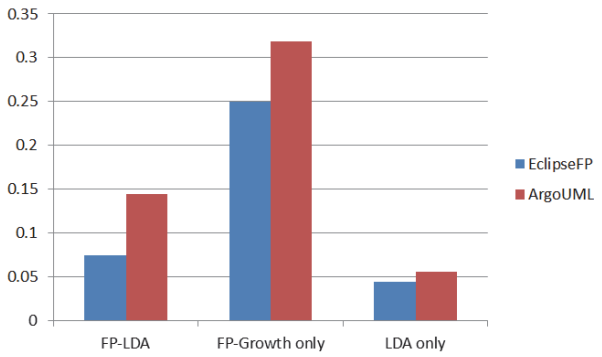


Figure 3. Average precision for precision at 15.

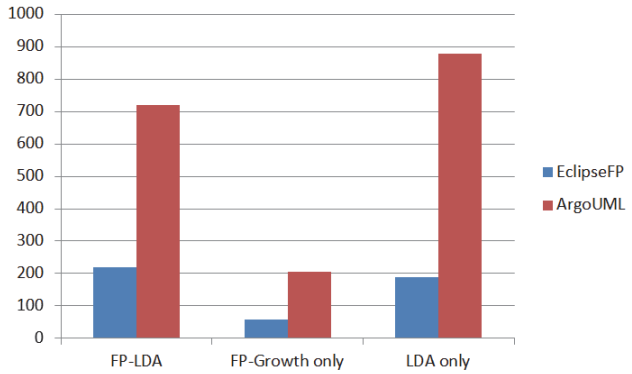


Figure 4. Number of files per transaction in the withheld transaction dataset that has at least one correct recommendation.

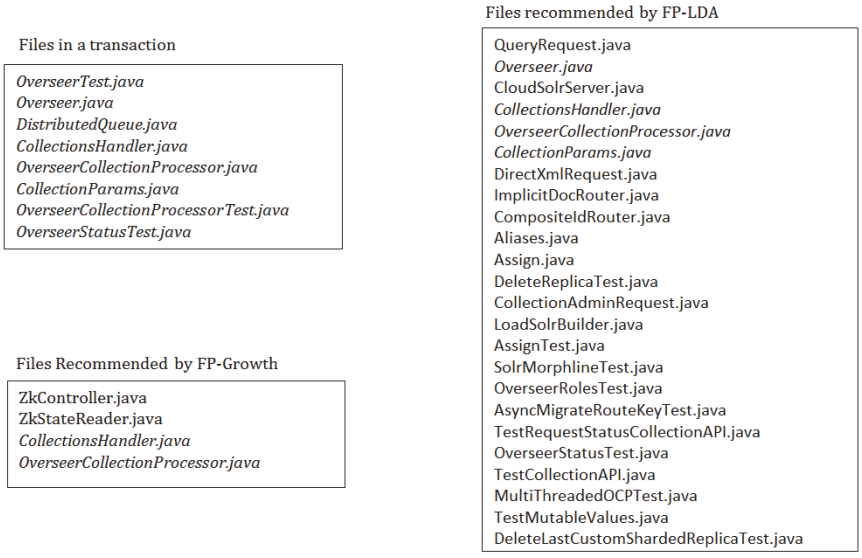


Figure 5. Example for Case 1: Recall for FP-Growth > 0 and FP-LDA recall > FP-Growth recall.

Case 2: FP-Growth recall =0, FP-LDA recall> 0: This study was done on ArgoUML revision ID 19876. Figure 6 shows the files that are checked in as part of this revision ID.

The minimum support and confidence values used for FP are 10% and 40%, respectively. The percentage cutoff distribution is 75%.

Using FP-Growth technique gives no recommendations therefore recall is 0. FP-LDA provides 11 recommendations. Four recommendations are correct for this transaction giving a recall of 0.5 and precision of 0.4. Figure 6 also shows the files recommended by FP-LDA.

D. Discussion

The calculation of precision and recall gives a general understanding of how the approach fares in finding relationships. We assumed that each of these transactions was a task presented to a developer. For each file in the test transaction, we calculated precision and recall values to see if the tool can predict the remaining files.

We have shown in the example for Case 2 a case where FP-Growth is not able to find file relationships, but FP-LDA overcomes this shortcoming. Because the number of total recommendations increases with topic modeling, the precision has a tendency to decrease, as shown in the example. However, FP-LDA does achieve the same precision rates for the same recall as FP-Growth alone. A higher recall value shows that there is an increase in the number of relevant files predicted. This indicates that number of correct recommendations increases with LDA. The utility of the approach lies in the fact that a developer needs to search only the set of recommended files, and not the entire source code base. Moreover, since we solely base our precision and recall on actual check-in records in the latter 10% of the history record, it is entirely possible that two files are related, but they may not have been checked-in together within this subset of the data, within the same transaction.

We have also shown in the experiment on two open source projects that FP-LDA is able to overcome the limitations of both FP-Growth and LDA only. FP-Growth can provide high precision rates but can recommend a very small number of files. Meanwhile, LDA only can recommend large number of files but with much lower precision. FP-LDA achieves a better balance between precision and the number of correct recommendations.

E. Limitations of the study

Our precision and recall numbers may be subject to the specific datasets we selected. However, since we selected projects with different characteristics and we observe the same trend across the different projects, this indicates that our results are applicable to other open source projects.

The number of topics used in LDA may also affect precision and recall rates. We ran our technique using different topic numbers and observed that the smaller the topic number, the higher the recall rates and the lower precision rates are generated. 50 and 100 topics are generally used by machine learning researchers. We added 25 topics to provide us a wider range of precision and recall values to examine. Also LDA may not always generate significant topics. The quality of generated topics has to be manually evaluated.

It is possible that our baseline, FP-Growth, could have produced more rules if we provided lower minimum support and confidence values. Choosing minimum support and minimum confidence for association mining is critical. A lower minimum support may yield more rules but not necessarily meaningful rules. Choosing the optimum value depends on the kind of dataset used. We decided the minimum support and confidence values based on our experience with analyzing projects and size of the version

Figure 6. Example for Case 2: Recall for FP-Growth = 0 and FP-LDA recall > 0.

history. However, even if more rules were produced, based on the examples shown, it is not possible for FP-Growth to predict certain files because the files modified in the most recent 10% of the history are not necessarily the same as the files modified in the first 90% of the history.

VI. LESSONS LEARNED

Lesson 1: Stop word selection. We observed that choice of the stop words affects the quality of topic model in a profound way. The words that need to be excluded from analysis depend largely on the use. For example, in this study we focused on finding the relationship between files. We do not want to know author file relationship. If we did not remove the author names from the processed source code files, we would get a topic with author names in it. In addition, the words we extracted for topic modeling represented different levels of abstraction. Thus, the highest level concepts (i.e., project-wide concepts) should also be added to the list of stop words. Since we are analyzing source code, the generally used list of stop words for natural language documents (e.g., a, an, the) are not applicable. At the same time, we wished to use a general approach for determining project-specific concepts that do not contribute to the meaning of each source code file. Thus, we used the following approach in creating our stop words list. We ran the corpus through a three-step process. First, we eliminated all language-specific reserved words. In Java, these include words such as "public", "class", "while". After the language-specific words are eliminated from the corpus, we then analyzed the corpus for the highest frequency words for that project [44]. We took the 10% of the highest frequently occurring words in the corpus and used these as our second set of stop words. Lastly, we examined the generated topics manually and removed any more words that does not contribute to the meaning of the topics (e.g., copyright info).

Lesson 2: Aggregating commits. In our previous work [1], we considered each atomic commit as one transaction. This time, we logically grouped transactions to obtain more meaningful changed sets. These heuristics are time interval and author. We assumed that the commits within a time interval by the same author are related to each other. After examining certain transactions we decided the time interval to be one hour. However, using a fixed time interval may not be generalizable across different projects. In the future, we plan to determine how to create a generalizable heuristics for aggregating commits to get more relevant results.

VII. CONCLUSION AND FUTURE WORK

In this paper, we used association mining and topic modeling together to assist developers in software maintenance task. These techniques were used to uncover the source file dependencies within a software project. We applied association mining on version history of a project to find files that frequently change together. We complemented this technique by using topic modeling on the source code documents. We showed that using topic modeling could uncover file dependencies that are not captured due to lack of version history for those files. Our evaluation indicates that this combination of techniques increases the number of relevant files obtained by at least a 100%, based on the seven open source projects we analyzed.

In the future, we would like to explore various options that can measure the usefulness of this approach. We plan to analyze more open source projects as well as conduct user studies to determine whether our approach reduces the time required for impact analysis or any maintenance task. In

Table II. Summary of precision and recall for the two cases examined.

Project	FP-Growth		FP-LDA		Topic Modeling Files Recommended
	Precision	Recall	Precision	Recall	
Case 1: FP-Growth has no recommendations while FP-LDA gives recommendations					
Argo UML	0	0	0.4	0.5	Revision id = 19876: 11 with 4/8 correct
Lucene	0.5	0.3	0.2	0.6	Revision Id = 1580463 24 with 4/8 correct
Case 2: FP-LDA has higher number of correct recommendations than FP-Growth					
Lucene	0.5	0.3	0.5	1	Revision Id = 1610028 9 with 0 correct

addition, we can use an approach to automatically rank the LDA topics based on their semantic importance to eliminate insignificant topics [45].

ACKNOWLEDGMENT

We thank Arthur U. Asuncion for his insights on LDA and providing the CVB0 implementation of LDA. We also thank Eamon Maguire for his assistance in extracting version histories and running the mining algorithm. We thank Delmar Davis with his assistance in running evaluations. We also thank Subha Vasudevan for her contributions, including analyzing generated topics and preparing the stop words for effective topic modeling. This material is based upon work supported by the National Science Foundation under Grant No. CCF-1218266. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] N. Dave, D. Davis, K. Potts, and H. U. Asuncion, "Uncovering file relationships using association mining and topic modeling," in *The Sixth International Conference on Information, Process, and Knowledge Management*, pp. 105–111, Mar 2014.

[2] S. S. Yau and J. S. Collofello, "Some stability measures for software maintenance," *Trans. on Software Engineering*, vol. SE-6, pp. 545–552, Nov. 1980. doi:10.1109/TSE.1980.234503.

[3] A. J. Ko, B. A. Myers, M. Coblenz, and H. H. Aung, "An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks," *TSE*, vol. 32, pp. 971–987, Dec 2006. doi:10.1109/TSE.2006.116.

[4] R. N. Taylor, N. Medvidovic, and E. Dashofy, *Software Architecture: Foundations, Theory, and Practice*. John Wiley & Sons, 2010.

[5] M. Sharp and A. Rountev, "Static analysis of object references in RMI-based Java software," in *Proc of the Int'l Conference on Software Maintenance*, pp. 101–110, Sep. 2005. doi:10.1109/ICSM.2005.84.

[6] M. Eaddy, A. V. Aho, G. Antoniol, and Y. G. Gueheneuc, "Cerberus: Tracing requirements to source code using information retrieval, dynamic analysis, and program analysis," in *Proc of the 16th Int'l Conference on Program Comprehension*, pp. 53–62, Jun. 2008. doi:10.1109/ICPC.2008.39.

[7] D. Cubranic, G. C. Murphy, J. Singer, and S. Booth Kellogg, "Hipikat: a project memory for software development," *Trans. on Software Engineering*, vol. 31, pp. 446–465, Jun. 2005. doi:10.1109/TSE.2005.71.

[8] M. P. Robillard, R. J. Walker, and T. Zimmermann, "Recommendation systems for software engineering," *IEEE Software*, vol. 27, pp. 80–86, Jul-Aug. 2010. doi:10.1109/MS.2009.161.

[9] S. Bajracharya, J. Ossher, and C. V. Lopes, "Sourcerer - an infrastructure for large-scale collection and analysis of open-source code," *Science of Computer Programming*, vol. 79, pp. 241–259, Jan. 2014. doi:10.1016/j.scico.2012.04.008.

[10] R. Agrawal and R. Srikant, "Fast algorithms for mining association rules in large databases," in *Proc of 20th Int'l Conference on Very Large Data Bases*, pp. 487–499, Sep. 1994.

[11] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation," in *Proc of the 2000 Int'l Conference on Mgmt of Data*, pp. 1–12, May 2000. doi:10.1145/342009.335372.

[12] D. M. Blei, "Probabilistic topic models," *Comunications of the ACM*, vol. 55, pp. 77–84, Apr 2012. doi:10.1145/2133806.2133826.

[13] R. Agrawal, T. Imielinski, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, pp. 207–216, Jun. 1993. doi:10.1145/170036.170072.

[14] A. T. T. Ying, G. C. Murphy, R. Ng, and M. C. Chu-Carroll, "Predicting source code changes by mining change history," *Trans. on Software Engineering*, vol. 30, pp. 574–586, Sep. 2004. doi:10.1109/TSE.2004.52.

[15] B. Liu, W. Hsu, and Y. Ma, "Mining association rules with multiple minimum supports," in *Proc of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining*, pp. 337–341, Aug. 1999. doi:10.1145/312129.312274.

[16] K. Lai and N. Cerpa, "Support vs confidence in association rule algorithms." http://www.researchgate.net/publication/233754781_Support_vs_Confidence_in_Association_Rule_Algorithms/file/9fcfd512a4907b8aca.pdf, 2001.

[17] B. Liu, W. Hsu, S. Chen, and Y. Ma, "Analyzing the subjective interestingness of association rules," *Intelligent Systems and their Applications*, vol. 15, pp. 47–55, 2000. doi:10.1109/5254.889106.

[18] J. Pei and et al., "Mining sequential patterns by pattern-growth: the PrefixSpan approach," *Trans. on Knowledge and Data Engineering*, vol. 16, pp. 1424–1440, Nov. 2004. doi:10.1109/TKDE.2004.77.

[19] C. Tjortjis, L. Sinos, and P. Layzell, "Facilitating program comprehension by mining association rules from source code," in *Proc of the International Workshop on Program Comprehension*, pp. 125–132, 2003. doi:10.1109/WPC.2003.1199196.

[20] T. Xie and J. Pei, "MAPO: mining API usages from open source repositories," in *Proc of the 2006 Int'l Workshop on Mining Software Repositories*, pp. pages 54–57, 2006. doi:10.1145/1137983.1137997.

[21] D. Rousidis and C. Tjortjis, "Clustering data retrieved from java source code to support software maintenance: A case study," in *Proc of the Ninth European Conference on Software Maintenance and Reengineering*, pp. 276–279, 2005. doi:10.1109/CSMR.2005.16.

[22] J. David, M. Koegel, H. Naughton, and J. Helming, "Traceability ReARMed," in *Proc of the International Computer Software and Applications Conference*, pp. 340–348, 2009. doi:10.1109/COMPSAC.2009.52.

[23] H. Kagdi, S. Yusuf, and J. I. Maletic, "Mining sequences of changed-files from version histories," in *Proc of the International Workshop on Mining Software Repositories*, pp. 47–53, 2006. doi:10.1145/1137983.1137996.

[24] B. Liu, W. Hsu, and Y. Ma, "Identifying non-actionable association rules," in *Proc of Int'l Conference on Knowledge Discovery and Data Mining*, pp. 329–334, Aug. 2001. doi:10.1145/502512.502560.

[25] T. Zimmermann, A. Zeller, P. Weissgerber, and S. Diehl, "Mining version histories to guide software changes," *Trans. on Software Engineering*, vol. 31, pp. 429–445, Jun. 2005. doi:10.1109/TSE.2005.72.

[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc of the 10th International Conference on World Wide Web*, pp. 285–295, May 2001. doi:10.1145/371920.372071.

[27] H. U. Asuncion, A. U. Asuncion, and R. N. Taylor, "Software traceability with topic modeling," in *Proc of the Int'l Conference on Software Engineering*, vol. 1, pp. 95–104, May 2010. doi:10.1145/1806799.1806817.

[28] S. P. Reiss, "Semantics-based code search," in *Proc of the Int'l Conference on Software Engineering*, pp. 243–253, May 2009. doi:10.1109/ICSE.2009.5070525.

[29] R. Oliveto, M. Gethers, D. Poshyvanyk, and A. De Lucia, "On the equivalence of information retrieval methods for automated traceability link recovery," in *Proc of the 16th Int'l Conference on Program Comprehension*, pp. 68–71, Jun-Jul. 2010. doi:10.1109/ICPC.2010.20.

[30] A. Asuncion, M. Welling, P. Smyth, and Y. W. Teh, "On smoothing and inference for topic models," in *Proc of Conference on Uncertainty in Artificial Intelligence*, pp. 27–34, Jun. 2009.

[31] B. Gretarsson and et al., "TopicNets: Visual analysis of large text corpora with topic modeling," *Trans. on Intelligent Systems and Technology*, vol. 3, pp. 23:1–23:26, Feb. 2012. doi:10.1145/2089094.2089099.

[32] E. Linstead, P. Rigor, S. Bajracharya, C. Lopes, and P. Baldi, "Mining Eclipse developer contributions via author-topic models," in *Proc of the Fourth International Workshop on Mining Software Repositories*, p. 30, 2007. doi:10.1109/MSR.2007.20.

[33] M. Gethers and D. Poshyvanyk, "Using relational topic models to capture coupling among classes in object-oriented software systems," in *Proc of the International Conference on Software Maintenance*, pp. 1–10, 2010. doi:10.1109/ICSM.2010.5609687.

[34] A. Nguyen, T. T. Nguyen, H. Nguyen, and T. N. Nguyen, "Multi-layered approach for recovering links between bug reports and fixes," in *Proc of the 20th International Symposium on the Foundations of Software Engineering*, p. 63, 2012.

[35] D. Han, C. Zhang, X. Fan, A. Hindle, K. Wong, and E. Stroulia, "Understanding Android fragmentation with topic analysis of vendor-specific bugs," in *Proc of the Working Conference on Reverse Engineering*, pp. 83–92, 2012. doi:10.1109/WCRE.2012.18.

[36] A. Panichella and et al., "How to effectively use topic models for software engineering tasks? an approach based on genetic algorithms," in *Proc of the Int'l Conference on Software Engineering*, pp. 522–531, May 2013. doi:10.1109/ICSE.2013.6606598.

[37] A. Nguyen, T. T. Nguyen, T. N. Nguyen, D. Lo, and C. Sun, "Duplicate bug report detection with a combination of information retrieval and topic modeling," in *Proc. of the International Conference on Automated Software Engineering*, pp. 70–79, 2012. doi:10.1145/2351676.2351687.

[38] T. Savage, B. Dit, M. Gethers, and D. Poshyvanyk, "TopicXP: Exploring topics in source code using Latent Dirichlet Allocation," in *Proc of the Int'l Conference on Software Maintenance*, pp. 1–6, Sep. 2010. doi:10.1109/ICSM.2010.5609654.

[39] TMate Software, "SVNKit." http://svnkit.com/. 2014.12.17.

[40] "JavaGit." http://javagit.sourceforge.net/. 2014.12.17.

[41] "ANTLR." http://www.antlr.org/. 2014.12.17.

[42] F. Coenen, G. Goulbourne, and P. Leng, "Tree structures for mining association rules," *Data Mining and Knowledge Discovery*, vol. 8, pp. 25–51, Jan. 2004. doi:10.1023/B:DAMI.0000005257.93780.3b.

[43] B. Li and L. Han, "Distance weighted cosine similarity measure for text classification," in *Intelligent Data Engineering and Automated Learning* (H. Yin, K. Tang, Y. Gao, F. Klawonn, M. Lee, T. Weise, B. Li, and X. Yao, eds.), vol. 8206 of *Lecture Notes in Computer Science*, pp. 611–618, Springer Berlin Heidelberg, 2013.

[44] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[45] L. Alsumait, D. Barbará, J. Gentle, and C. Domeniconi, "Topic significance ranking of LDA generative models," in *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases: Part I*, pp. 67–82, 2009.

Smartphone Based 3D Navigation Techniques in an Astronomical Observatory Context: Implementation and Evaluation in a Software Platform

Louis-Pierre Bergé[1], Gary Perelman[1], Adrien Hamelin[1], Mathieu Raynal[1], Cédric Sanza[1], Minica Houry-Panchetti[1], Rémi Cabanac[2], and Emmanuel Dubois[1]

[1] University of Toulouse
IRIT
Toulouse, France
{Louis-Pierre.Berge, Emmanuel.Dubois}@irit.fr

[2] Télescope Bernard Lyot, Observatoire Midi-Pyrénées
University of Toulouse
Tarbes, France
remi.cabanac@obs-mip.fr

***Abstract*—3D Virtual Environments (3DVE) come up as a good solution to transmit knowledge in a museum exhibit. In such contexts, providing easy to learn and to use interaction techniques which facilitate the handling inside a 3DVE is crucial to maximize the knowledge transfer. We took the opportunity to design and implement a software platform for explaining the behavior of the Telescope Bernard-Lyot to museum visitors on top of the Pic du Midi. Beyond the popularization of a complex scientific equipment, this platform constitutes an open software environment to easily plug different 3D interaction techniques. Recently, popular use of a smartphones as personal handled computer lets us envision the use of a mobile device as an interaction support with these 3DVE. Accordingly, we design and propose how to use the smartphone as a tangible object to navigate inside a 3DVE. In order to prove the interest in the use of smartphones, we compare our solution with available solutions: keyboard-mouse and 3D mouse. User experiments confirmed our hypothesis and particularly emphasizes that visitors find our solution more attractive and stimulating. Finally, we illustrate the benefits of our software framework by plugging alternative interaction techniques for supporting selection and manipulation task in 3D.**

Keywords-museum exhibit, 3D environment, software platform, interactive visualization, interaction with smartphone, 3D navigation, experiment.

I. INTRODUCTION

In comparison to our initial work [1], this paper explored the context of astronomical observatory in museum exhibit, integrating our previous experiment and creating a larger software platform for public demonstrations. Nowadays, 3D Virtual Environments (3DVE) are no longer restricted to industrial use and they are now available to the mass-market in various situations: for leisure in video games, to explore a city on Google Earth or in public displays [2], to design a kitchen on a store website or to observe 3D reconstructed historical sites on online virtual museum [3]. The use of a 3DVE in a museum, to engage people and facilitate the transmission of knowledge by adding fun and modern aspect [4], is the core of this work. As part of collaboration with the Telescope Bernard-Lyot (TBL), we intended to provide a support for helping visitors to better understand how this telescope operates. Indeed, access to the real telescope is restricted to keep the best possible values of temperature and hygrometry, crucial parameters for collecting good observation results. The telescope only works at night and thus cannot be observed in a running context during the day. Therefore, an interactive application support was required to help visitors discover the TBL. It has been shown that 3D virtual environments can help understand a complex system by improving the spatial representation users have of the objects or by allowing cooperation between different users [5]. Therefore, we developed an interactive 3D environment representing the TBL and its behavior.

However, in a museum context, the visitor's attention must be focused on the content of the message and not distracted by any difficulties caused by the use of a complex interaction technique. This is especially true in a museum where the maximization of the knowledge transfer is the primary goal of an interactive 3D experience. Common devices, such as keyboard and mouse [6] or joystick [7] are therefore widely used in museums. To increase the immersion of the user, solutions combining multiple screens or cave-like devices [8] also exist. However, these solutions are cumbersome, expensive and not widespread.

Alternatively, the use of a smartphone, as a personal handheld computer, is commonly and largely accepted. Smartphones provide a rich set of features and sensors that can be useful to interact, especially with 3DVE and with remote, shared and large displays. Smartphones also create the opportunity for the simultaneous presence of a private space of interaction and a private space of viewing coupled with a public viewing on another screen. Furthermore, many researches have already been performed with smartphones to study their use for interacting with a computer. They explore multiple aspects such as technological capabilities [9], tactile interaction techniques [10], near or around interaction techniques [11]. Given the potential in terms of interaction support and the availability of smartphones in anyone's pocket, we explore the benefits and limitations of the use of a smartphone for interacting with a 3DVE displayed in a museum context in this article.

Beyond the development of an interactive 3D environment for science popularization with only one set of interaction techniques, we took the opportunity to design and implement a platform on which it is easy to plug interaction techniques of different forms. The contribution presented in this paper is thus threefold.

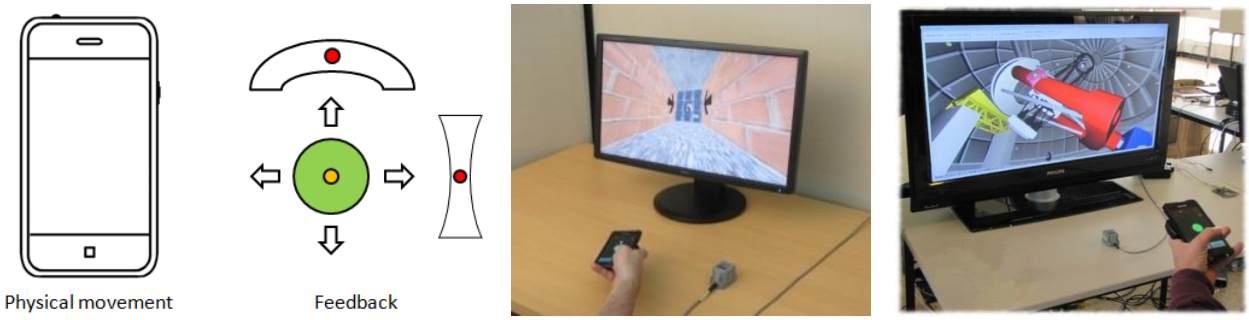


Figure 1. Smartphone-Based interaction techniques to navigate into a 3D museum exhibit: a) principle of use, b) controlled experiment, c) integration in an in-situ exhibit.

First, we adopt a user-centered approach to address the concrete need of the Telescope Bernard-Lyot: providing a 3D interactive platform for museum exhibit, which facilitates the understanding of the telescope behavior.

Second, we describe the design and implementation of the interactive platform. The platform is based on an open architecture in order to be able to plug different interaction modalities easily. Fundamental features integrated in the platform are then presented, such as 3D rendering, data visualization and interaction based on widows, mouse and 3D mouse device. As a result, this platform provides a pedagogical environment in which different interaction techniques can easily be plugged into one specific 3D environment: the objects, the size of the elements and the available actions are the same for every technique.

Third, we design and evaluate a new smartphone-based technique for navigating inside the 3D interactive platform: indeed navigation is the most predominant task a user will have to perform in order to discover and understand the virtual space. Concretely, our technique translates the motions of the smartphone into motions of the point of view in the 3DVE. We thus propose 1) to consider a smartphone as a tangible object, in order to integrate it smoothly into a museum environment and because it has been proven to be easier to apprehend by newcomers [12], 2) to display feedback and/or personalized information on the smartphone display, 3) to deport the display of the 3DVE on a large screen, in order to provide a display space visible by multiple users as required in museum contexts. As a result, our technique combines the use of a popular and personal portable device, the physical space surrounding the device and the user's gestures and input for navigating inside a 3DVE. We compared our designed solution to the use of more common and available technologies: the keyboard-mouse device and a 3D mouse device. We proposed a controlled evaluation focused on the interaction techniques: out of a specific museum context, the user will not be distracted by pedagogical content. We measure its usability and attractiveness in conjunction with performance considerations. The results confirm the interest of considering the use of personal mobile devices for navigating inside a 3DVE: the results are particularly significant in terms of a user's attractivity.

The paper is structured as follow. We first position our work with regards to the context of 3D interactive application in museums (Section II). Then we introduce our design approach of the platform and the relevant software considerations related to the platform implementation in Section III. After, we detail our smartphone-based interaction technique (Section IV) with the settings of the user experiment (Section V) and the results (Section VI). We finally discuss (Section VII) the validation of our 3D interactive platform, the originality and limitations of our smartphone-based solution and other interaction techniques integrated in our platform (Section VIII). We conclude with perspectives and future works.

II. Building a 3D Interactive Platform for Museum Exhibit: Related Works

As already mentioned, interactive 3D virtual environments are emerging in public space applications and especially in museum contexts where they contribute to the immersion and engagement of visitors.

In this field, many works have already been done and we propose in this section a review of already well known and explored challenges. First, we focus on the different forms a 3D interactive visiting application can take in a museum. Then we synthesize technological considerations of importance and finally look at the design approaches for building a 3D interactive virtual exhibit.

A. Visiting assistance

In order to enhance the visit in museums, tour guides or audio-guide are available. These guides provide information about the different sections of a museum and guide the user through they visit by defining a specific path to follow. In the context of 3D, this raises specific challenges. First, the application must help the visitor so that he does not get lost in the 3D environment. Offering landmarks, maps, etc. are thus required [13]. The realism of the scene also plays a role [14]. To complement this immersion, avatars, conversational agents, realistic humanoids are also inserted in such app to guide or provide explanations to the visitors [15].

Alternatively, a smooth mix of real and virtual enables the visitor to proceed through a real museum, to collect elements of interests, and after the visit to virtually re-explore in 3D those collected elements [16]. This is close to the attempts of using 3D virtual environments to support post-visit of a museum [17].

Beyond those individual visits, 3D virtual visits also raise the possibility to develop true collaborative visits so that remote visitors can visit it together [18]. To some extent, it is also a possible solution to explore in 3D some of the objects exposed in the museum. The concept of collaborative exhibit is especially relevant for blind visitors [19].

B. Technological considerations

To settle these different forms of 3D interactive museum visits, several technical issues are raised and addressed in the scientific literature. In the museum context, two main considerations emerge from the literature: the 3D environment realism and the remote access.

1) 3D use and constraints

Although not specific to museum contexts, many ways of modeling museum objects into a 3D scene have been explored. It ranges from the modeling of the objects using dedicated approaches [20] to really precise modeling with a 3D laser scanner [21] through image based reconstruction [22]. In addition, the rendering of the resulting 3D meshes can be crucial in order to give the user a good experience during a virtual visit: it involves a large graphical management as the handling of light effects on the objects or realistic shadows [23]. More complex treatments may also be required to obtain high resolution, accurate and complete 3D models of parts of an exhibit. This is for example possible through the combination of different 3D model types [24].

2) Collaborative access, distant access

Once the virtual exhibit has been built, it remains necessary to allow its use in the context of a museum. Specific software design aspect must be considered to provide a support to an effective distant and collaborative access [25]. Indeed, as many visitors may come at the same time to the museum section, they have to be able to interact with the environment at the same time without having to wait the other visitors to leave. Many software architectures have been defined in the scientific literature to deal with such problems [26]. For example, if the 3D virtual environment allows a visitor to manipulate an element of the virtual museum, control over the environment must be handled by the platform. Apart from the collaborative or simultaneous access, visitors may have access to the virtual exhibit from everywhere: to this end, web access has been explored [14].

C. Building support to virtual exhibits

Behind the form, rendering and software considerations, developing a virtual exhibit also requires to think of the structure of the application. This must be done in two ways.

1) Management systems

The first way consists in structuring the development of the application. Tool chains to create, store, manage and visualize all the data needed in virtual exhibits have been defined [27]. Other approaches have also been proposed where non-familiarized-in-computer-science users can edit their own application [27]. Such systems help the museum staff to manage their virtual museum easily and allow dynamic modifications (adding or removing objects, adding museum sections that does not exists in the real museum, etc.) [27].

2) Learning perspectives

The second way consists in structuring the knowledge transfer approach. Indeed, once the 3D environment has been built, there is still the need to consider the ultimate goal of this virtual environment: its knowledge transfer capabilities. Some researchers have been led to determine if the use of a 3D environment helps the user in the learning process [4]. After experiments with several museums with or without a 3D virtual environment, it has been shown that these virtual accesses allow a better understanding of the museum information. This finding is based on the fact that the users are not forced to follow a specific path during their visit and thus are more concentrated on the museum section they are visiting.

3D virtual spaces also offer tools and support for a pedagogical transfer. The impact of different tools is studied to reveal how it may enhance the learning experience [28].

D. Interaction techniques for 3D museum applications

Finally, the fourth major areas of research related to interactive 3D virtual environment in museum is focusing on the user's interaction with and within the 3D environment. Many interaction techniques dedicated to a 3D virtual museum environment have been studied in the past few years. Indeed, interacting with a 3D virtual exhibit application can be achieved using a simple GUI interface [29], where each object is represented by a tab that can be activated. It is possible to use tactile devices [30] to allow zooming on particular objects or turning them around to see them on a different point of view. Haptic devices can also be used [8] to enhance the spatial knowledge of the users when they manipulate the 3D objects of the application. Finally, augmented reality can be used to upgrade the immersive capabilities of the virtual exhibit [31].

E. Outcome of the state of the art

According to the scientific literature, the use of 3D in a museum context has been widely addressed over the last years. Many different preoccupations have been raised and considered such as the integration of such application in a museum, 3D rendering issues and interaction techniques.

Technological advances now largely allow the use of various and advanced forms of interactions. It is therefore required to provide a way to change interaction techniques easily in order to support the opportunity offered by new or emerging forms of interaction.

Furthermore, little attention has been paid so far to the use of personal devices for navigating 3D in a museum context. Exploring the potential support personal smartphone can offer to perform complex interaction such as tasks in 3D environment are thus also required.

In our work, the visiting assistance we are interested in is limited to a virtual interactive application, used in a museum with museum mediators available to explain the information provided by the application or using the application to catch an audience. From a technological point of view, we are seeking to provide a support to easily test different interaction techniques. Therefore, we need a software platform allowing easy plug and play of multiple interaction

techniques. Regarding the application supporting the visit, we are focusing on a way to disseminate information about the Telescope Bernard Lyot (TBL) to none experts of astronomy and telescopes. We are more specifically focusing on the understanding of the operators' activity on the TBL. Finally, in terms of interaction techniques we are mainly interested in providing a support for individual visitor to use a smartphone device to perceive a detailed area of one part of the 3D scene rendered on the remote and large screen.

From these considerations, we designed, implemented and evaluated a set of interaction techniques based on the use of a smartphone for 3D virtual museum exhibit. The different interaction techniques have been plugged into a new platform that handles a 3D virtual museum exhibit based on a user-centered design approach. The aim of the platform is also to contribute to better take into account the user when developing a 3D interactive museum exhibit by offering a support to easily plug new interaction techniques: our goal behind the platform is thus to plug and compare different interaction techniques in a scenario and a 3D environment that serve as references. As a consequence, the platform architecture, which has been designed according to the state of art, also contains particular features that allow a better handling of the multiple interaction techniques that can be plugged on it.

III. DESIGNING THE 3D INTERACTIVE PLATFORM

As already mentioned, the platform is simultaneously intended to be a support for explaining how a telescope works and easy to plug interaction techniques for museum 3D virtual environments.

The user-centered design approach we followed to reach these two goals includes two steps. It firstly includes an observation step in which the goal is to identify the requirements for the interactive exhibit of the TBL. The design software architecture and the interaction techniques have been led in a second place. In the following sections we report the design tools used along the different steps and the design decisions taken to develop the platform.

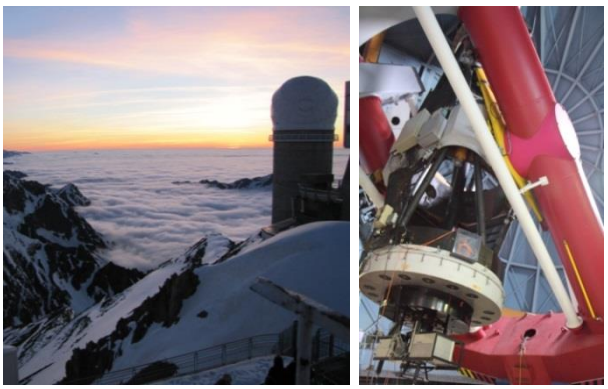


Figure 2. The Telescope Bernard-Lyot outside (left) and inside (right).

A. Context

The Telescope Bernard-Lyot (TBL) is located on top of the Pic-du-Midi (Figure 2). It is the largest telescope in Europe (2m.) and it takes advantage of a very good position: high altitude, far from city lights, etc. The operators of the TBL are in charge of planning and executing observations of "stars". These observations are requested by European researchers.

To perform these observations, a Windows, Icons, Menus and Pointing device (WIMP) application allows the monitoring of relevant data and the emission of "command-lines" to position the telescope and its dome. Several screens are displaying all the required information in a very basic form as illustrated on the Figure 3.

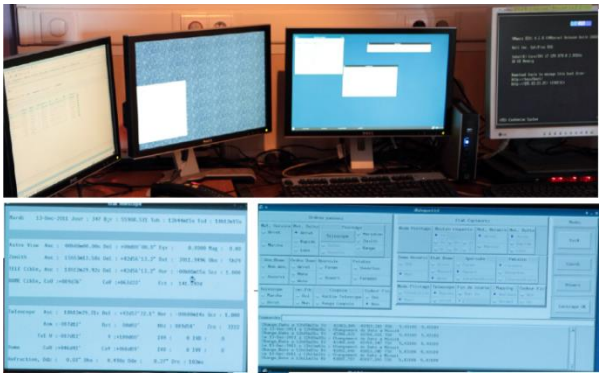


Figure 3. Current operators' desktop (top) and their WIMP applications (bottom).

B. Domain analysis

In this context, we performed two different forms of analysis: an in-situ observation to identify the tasks and data used by the current operators, and semi-guided interviews to define the potential end-users and the scenario describing the use of the TBL.

During the in-situ observation, we watched the operators of the TBL during their work in the supervision room. We also visited the dome to have a better representation of the telescope and its components. We discussed with them to understand the tasks they performed more finely and we observed the current WIMP application they used. We formalized this task analysis through a task tree expressed with K-MADe [32]. The modeling of the overall activity leads to the identification of approximately 60 tasks. This provides a representation of the operator's activity and the required data can easily be linked to the appropriate subtask. It constitutes a good support to organize the data retrieved in this first step of the analysis.

After that, we organized semi-guided interviews with the operators and the personnel in charge of the visits (at the existing museum and at the dome) to define their additional or more specific needs that did not appear in the first observations. These multidisciplinary and participative meetings were especially important to identify the audience and the most important steps of the activity of an operator that must be explained to the different audiences.

From this domain analysis step, we established a list of data that are the most relevant when performing an observation with the telescope:

- The two angles of the TBL
- The position of the seal of the dome (two angles).
- Inside and outside temperatures and hygrometry are also required to ensure a good quality observation.

We also ended with two potential targets, sharing a common scenario:

- A museum visitor
- An astronomy student who learns how such telescopes work.

Out of the task analysis we built with animators of the museum a defined pedagogical scenario allowing a random person to understand the set of operations performed to operate the telescope. It is split in five different steps, which globally represent how the telescope works and gathers data from a selected star:

- Activation of the hydraulic system
- Selection of a star
- Alignment of the telescope and the dome in the star direction
- Activation of the earth rotation and movement of the telescope to follow the star movement
- View of the gathered data

These steps are further detailed and illustrated in Section III.C.3. Knowing all these informations, we were able to develop a platform that not only answers the required features, but also provides an easy way to plug the interaction techniques of different modalities and connect them to the 3D environment.

C. Design

Obviously, the basis of the 3D interactive platform is the 3D scene. We thus started by building this environment on the basis of 3D meshes provided by the Telescope Bernard-Lyot. These meshes had been built in SketchUp [33] on the basis of the 2D blueprints of the TBL. Our 3D environment has been built using the open source 3D renderer: Irrlicht [34].

Beyond the 3D environment, the design of the platform was decomposed into three subparts: the software and architecture of the system, a first set of interaction techniques and the instantiation of the pedagogical scenario.

1) Software design

The software design must allow plugging different interaction techniques to be used and evaluated. The software perspective must also take into account the context of use of the platform: several museum visitors may interact simultaneously from different devices or even places, time constraints for the visit exists, etc. This resulted into three design considerations: the architecture, the multi-users management and the components communication.

To ensure a clear decomposition of the interaction and system parts, and allow for a modular implementation, an adaptation of the MVC architecture model is used to structure our platform.

The concept of *Model* is used in our case to handle the data related to the operated telescope: position of the different elements, temperature, humidity, etc. The TBL being only operated at night, a simulation is required at day time to demonstrate its behavior to visitors. Therefore, the concept of *Model* applied to our platform allows a connection to the real TBL and/or a simulation of it. The first *Model*, called "real telescope" (Figure 4), gathers the data of the TBL and transfers it to the platform. The second one, called "simulated telescope" (Figure 4), emulates the behavior of the TBL. Actually, the 3D software platform allows visitors to manipulate only the simulated telescope and visualize the real one.

The need to visualize the telescope data with a WIMP interface for example can directly be mapped to the concepts of *View* of the MVC pattern (Figure 4). Moreover, the need to manipulate the telescope or to navigate inside the 3D view can directly be mapped to the concepts of *Controller* of the MVC pattern (Figure 4). The 3D view is a mix of *View* and *Controller* concepts. In fact, the 3D view allows a visualization of the telescope data and also permits to manipulation element that control the scenario of the software platform (see Section III.C.3).

In the middle of these components, a "Dialog Controller" orchestrates these components and their synchronization (Figure 4). It also includes a waiting queue to dispatch the control over the multi-user setting. This software implementation supports an easy reconfiguration of the platform to fit with different types of tasks, settings, features and interaction techniques.

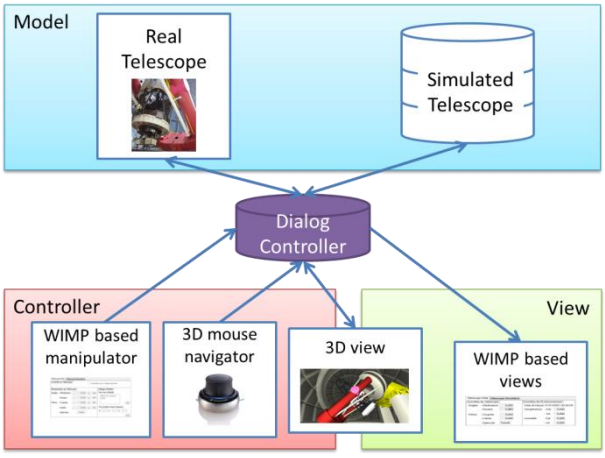


Figure 4. MVC architecture with different plugged components.

Finally, to ensure an easy-to-plug, multi-platform compliant architecture and multi-programming language, we based the software communication protocol on the Ivy network API [35]. We created a set of textual predefined messages to support the communication over the different components types. These messages are generic for plugging

different interaction technique into our software platform. As an example, *controller* components supporting manipulation of the 3D virtual environment has specific network messages to take control over the 3D scene and to control the telescope; *view* components have messages to update information regarding the telescope data modification. Those communications are encapsulated into DLLs in order to help future interaction technique developer to worry only about the interaction technique and not about their link to the platform.

2) Interaction techniques design

The second perspective of the design focuses on the way the users will concretely act on the 3D environments and its features. Users' interaction design is more specifically dedicated to the definition of a concrete interactive application to support the explanation and popularization of the scientific equipment. It aims at offering flexibility to the user in terms of access to the interactive system, i.e., to provide multiple interaction types to perform the three tasks we defined previously in the 3D environment.

We first lead a participatory design session, involving telescope operators, visitors, HCI specialists, 3D specialists and museum facilitators to produce mockups describing the organization and representation of the data required when operating the telescope. Results permits to design two different forms of data visualization and two sets of interaction technique.

a) Data visualization

Two forms of visualization are available on the current platform: a textual view and a 3D view of the data. The textual view is made of a set of labels with information such as the angles of the telescope and the position of the seal on the dome (Figure 5-left). It also contains hydrometric and temperature data, inside and outside the dome (Figure 5-right).

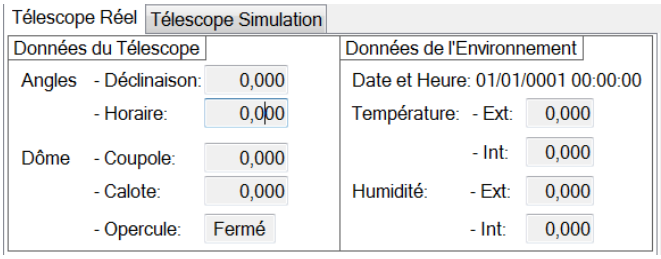


Figure 5. WIMP based view of the telescope data.

The 3D view provides a graphical representation of the current position and orientation of the elements of the telescope in 3D (Figure 6). Relevant data are also labeled where appropriate: angle values are displayed on the rotation axis, temperature is a flying label in the always visible dome, etc. The 3D view thus aggregates information in a single graphical and a realistic fairly representation.

Further, on-going iterations will optimize the presentation of the data. For example, recent participatory design sessions produced mock-ups for iconic representations of the labels: once developed they will be used to replace the textual label in the 2D view.

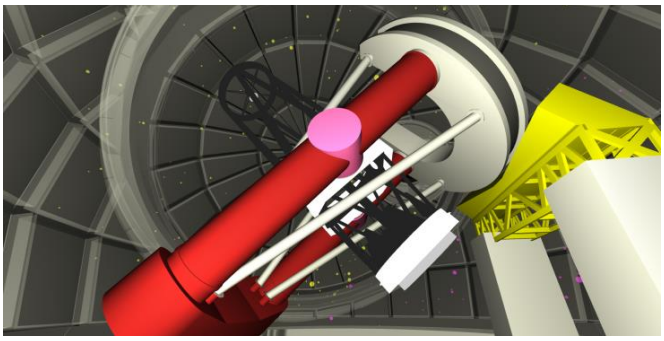


Figure 6. 3D view of the telescope.

b) Keyboard-mouse interaction technique

To cover all these manipulations, the simplest solution consists in offering the user traditional windows and buttons interface. This also constitutes the most realistic interaction with regard to the actual settings used by the real operators of the TBL (Figure 3). Figure 7-left show our WIMP interface to manipulate the elements of the telescope and to select a star from a list. In comparison to the initial WIMP interface, our domain analysis permits to limit the number of buttons and therefore simplify the interface for non-expert users. A similar interface allows controlling the earth rotation and thus observing the stars movement in the 3D view.

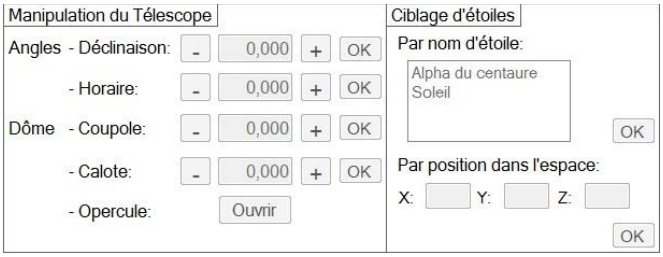


Figure 7. WIMP based telescope manipulator.

To complement this widget set and allow navigation in the 3D environment we also developed two alternatives. The first one involves the use of the mouse and keyboard, which are very common in video games. The keyboards arrow keys are used to perform translations in the environment and the mouse motions allow the user to orient its point of view in the scene. The second is a WIMP interface for different buttons for controlling the position and orientation in the 3D environment (Figure 8).

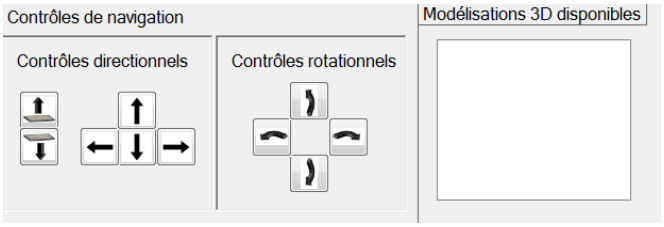


Figure 8. WIMP interface to navigate inside the 3DVE.

c) 3D mouse interaction technique

We have also integrated a commercial device dedicated to navigate inside the 3D environment: a 3D mouse, the Space Navigator [36] (Figure 9-left corner). This 3D mouse can be used to control the three rotation axis and three translation axes. A WIMP interface permits to calibrate and select the sensibility of the translation and rotation movement (Figure 9).

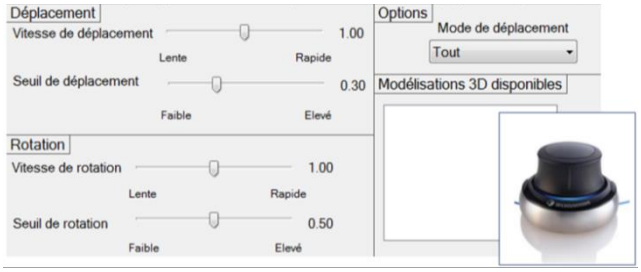


Figure 9. WIMP interface to configure the 3D mouse named Space Navigator [36] (left corner).

3) Instantiating the scenario on the platform

The third and last perspective clearly addresses the need to develop a concrete application to explain the behavior of the Telescope Bernard-Lyot through a scenario.

Given the 3D environment and the interaction techniques we developed, the five steps of the pedagogical scenario defined in close collaboration with the TBL and museum facilitators (see Section III.B) has been instantiated as follow.

The first step, "activation of the hydraulic system" consists in finding a red button placed in the 3D environment. This navigation task can be achieved using a WIMP interface or a 3D mouse navigator. Coming close to the button triggers a pressure on it. This button, although it does not match any button on the real telescope, represents the first step that the operators do when they use the telescope: initiating the equipment.

The second step, "selection of a star", consists in selecting the star to observe. To do this, the dome of the 3D environment fades and the user can see all the stars currently added in the 3D scene. Selecting a star can be done either by clicking on it thanks to the mouse in the 3D scene or by choosing a star in the WIMP interface (Figure 7-left).

The third step, "alignment of the telescope and the dome in the star direction", consists in manipulating all the elements of the 3D scene in order to observe the star. To do that, the user can move the 3D objects by using the WIMP interface (Figure 7-left). Alternatively, an automatic resolution can be executed by the platform to perform the appropriate translations and rotations of the different elements of the TBL.

The fourth step, "activation of the earth rotation and movement of the telescope to follow the star movement", emulates the earth rotation around its axis. From a human point of view, it seems that the stars are rotating in the sky around the polar star. As a real observation can last for hours, it is important to perceive the impact of the earth rotation on the work of the operator. The sky rotation in the 3D environment is of course a lot faster than in reality. The stars start rotating and the user has to constantly align the telescope in the star direction. Another interest of this step is to underline that the telescope has been built in such a way that there is only one axis that changes over time.

The fifth and last step, "view of the gathered data", shows to the user the data collected from the star observation performed through the previous steps (Figure 10). These data were gathered on real stars by the Telescope Bernard-Lyot and are shown to the user with an explanatory text. The data gathered are the spectrum of the star light obtained with a spectropolarimeter, one scientific originality of the equipment offered at the TBL.

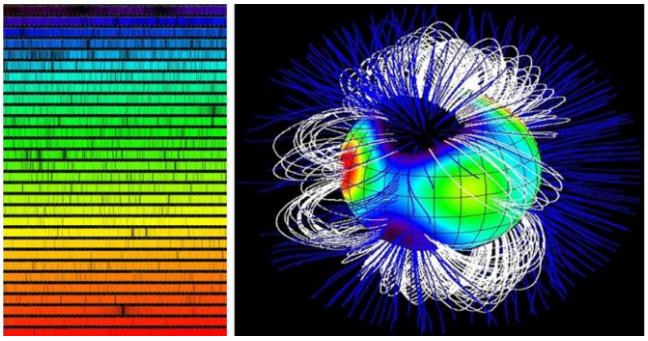


Figure 10. Spectrum of a star light (left) and its associated magnetic field (right).

As illustrated here, the set of interaction techniques, based on windows, label, keyboard-mouse and 3D mouse is well suited for professional contexts and for users who are familiar with the use of 3D environments. But targeted audience is also made of museum visitors who may be very occasional users of 3D. To better engage the visitor and to explore the adequacy of other forms of interactive techniques, we have therefore designed a second type of interaction techniques to navigate inside the 3D environment based on the use of a smartphone. In the next section, we will detail our interaction technique.

IV. OUR SMARTPHONE-BASED INTERACTION TECHNIQUE

As described in the introduction, our interaction technique is based on the handling, manipulation and use of a smartphone, a familiar and personal object for most users. Three major characteristics define our interaction technique: tangible manipulation of the smartphone, personalized data displayed on the smartphone and 3DVE displayed on a remote screen.

We restricted the degrees of freedom of the navigation task in order to be close to human behavior and existing solutions in video games with standard device: two degrees of freedom (DOF) are used for translations (front/back and left/right) and two for rotations (up/down and left/right). We did not include the y-axis translation and the z-axis rotation since they are not commonly used for the navigation task. To identify how to map the tangible use of the smartphone to these DOF, we first performed a guessability study as performed by other work [37].

A. Guessability study

14 participants have been involved and they were all handling their own smartphone in the right hand. To facilitate the understanding, the guessability study dealt with only one translation and one rotation. A picture of a 3DVE was presented to the participants on a vertical support (Figure 11-left). It included a door on the left of the 3D scene and the participants were asked to perform any actions they wished on their smartphones in order to be able to look through the door. A second picture (Figure 11-right) was then displayed: now facing the door, the participants were instructed to pass through the door.

Figure 11. Pictures presented during the guessability study.

In this second question, 11 participants performed hand translations to translate the point of view. Interestingly none suggested using the tactile modality. Results are more contrasted with the first question, requiring a rotation: 5 participants used a heading rotation of the handled smartphone; only one used the roll technique; three proposed to hit the target with their smartphone; five participants placed the smartphone vertically (either in landscape or portrait orientation) and then rotated the smartphone according to the vertical axis (roll) thus preventing the view on the smartphone screen.

B. Design solution

From the guessability study, we retained that physical translations of the smartphone seem to be the most direct way to perform translations of the point of view in the 3DVE. It has been implemented in our technique as follows. Bringing the smartphone to the left / right / front or back from its initial position triggers a corresponding shifting movement of the point of view in the 3DVE (Figure 12-a). The position of the point of view is thus controlled through a rate control approach; the applied rate is always the same and constant. It is of course possible to combine front / back translation with right / left translation.

Feedback is displayed on the smartphone while moving it to perform these translations (Figure 12-b). A large circle displayed on the smartphone represents the initial position of the smartphone and the physical area in which no action will be triggered: the neutral zone. A small circle represents the current position of the smartphone and arrows express the action triggered in the 3DVE. As long as the small circle is inside the large circle, the navigation in the 3DVE is not activated. The feedback provided during each of four possible motions is illustrated in Figure 12-b. Finally, the smartphone vibrates every time that the navigation action is changed.

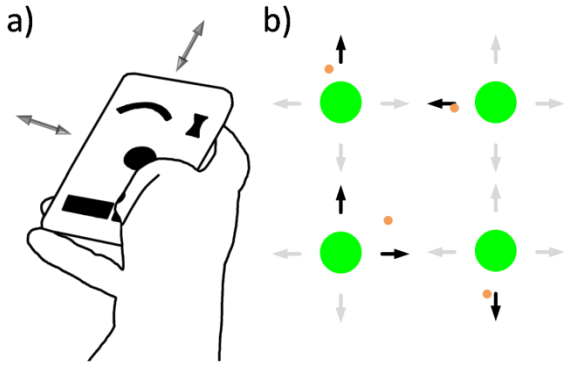


Figure 12. Our smartphone based interaction. (a) Physical action for translation. (b) Feedback of the translation: front, left, front and right, back translation.

Regarding rotations, we retain from the guessability study the most usable solution: rotations of the hand-wrist handling the smartphone are mapped to orientations of the point of view in the 3DVE. In our implementation, horizontal wrist rotations to the left/right of the arm are mapped to left/right rotations of the viewpoint (heading axis, rY) and wrist rotations above/below the arm are mapped to up/down rotations of the viewpoint (pitch axis, rX) (Figure 13-a). A position control approach has been adopted here that establishes a direct coupling of the wrist angle with the point of view orientation. A constant gain has been set for the wrist rotations: the limited range of 10° left and right [38] can be used to cover the range of the rotation angle inside the 3DVE (180°). This solution does not bear a U-turn: this was not required in the experiment but could be solved by transforming the position control into a rate control when the wrist reaches a certain angle.

As for translations, feedback is displayed on the smartphone while moving it to perform these rotations. It is rendered through two "spirit levels": they provide an estimation of the current orientations of the smartphone (Figure 13-b) with respect to the initial orientations used as a reference.

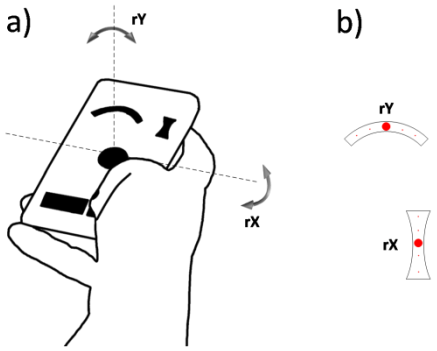


Figure 13. Our smartphone based interaction. (a) Physical action for rotation. (b) Feedback of the rotation.

To avoid unintended motions of the virtual camera in the 3DVE, the translations and rotations of the smartphone are applied to the 3DVE only when the user is pressing the button “navigate” displayed on the smartphone.

The smartphone also displays a “calibrate” button. This allows the user to recalibrate the smartphone at will, i.e., to reset the center of the neutral zone to the current position of the smartphone and the reference orientations.

Figure 14 illustrates the use of this smartphone-based interaction technique. Circles in the middle of the smartphone screen indicate that the user is in the neutral zone. Spirit levels on top of the smartphone screen indicate he is looking a little bit upwards (right spirit level) and slightly to the right (left spirit level). As the user's thumb is not pressing the "navigate" button displayed on the smartphone screen (left corner), the motions of the smartphone do not currently affect the point of view on the scene.

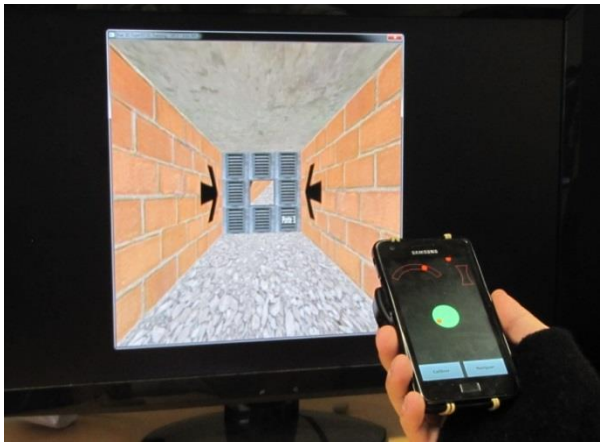


Figure 14. User navigating with the smartphone-based technique.

V. EXPERIMENT

We conducted an experiment to compare our smartphone-based interaction technique with two other techniques using devices available in museums: a keyboard-mouse combination and a 3D mouse. In the museum context, the temporal performances are not predominant. In fact our goal was to assess and compare the usability and attractiveness of these three techniques. Our protocol does not include museum information in order to keep the participant focused on the interaction task.

A. Task

The task consisted in navigating inside a 3D tunnel composed by linear segments ending in a door (Figure 15-b). The task is similar to the one presented in [39] and sufficiently generic to evaluate the interaction techniques correctly. The participants had to go through the segments and go across the doors but could not get out of the tunnel. Black arrows on the wall allowed finding the direction of the next door easily. The segments between the doors formed the tunnel, its orientation was randomly generated: the center of each door is placed -40, -20, 0, 20 or 40 pixels to the left/ right axis and to the top/bottom axis of the center of the previous door (Figure 15-a). One trial of the task consist in navigate inside a tunnel including all 25 possible directions of the next door. The movement of the user is not liable to gravity. When the user looks up and starts a front translation movement, the resulting motion is a translation in the direction of the targeted point.

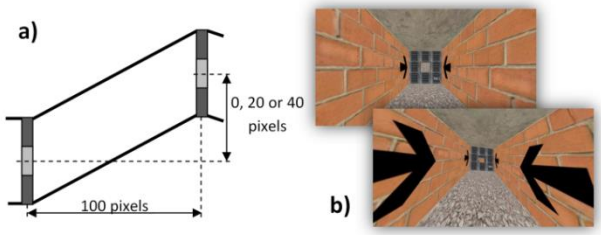


Figure 15. (a) Representation of one segment of the 3D tunnel. (b) Screenshots of the 3D environment of the experiment.

B. Interaction techniques

We compared three techniques: keyboard-mouse, 3D mouse and our technique based on a smartphone. In keyboard-mouse, the movements of the mouse control the 2 DOF point of view of the virtual camera (orientation). The four directional arrows of the keyboard control the 2 DOF of the translation of the virtual camera. In 3D mouse, the participant applies lateral forces onto the device to control translations (right/left, front/back), and rotational forces to control orientations of the virtual camera. The use of our technique, the smartphone, has been described in Section IV. For the three techniques, it appears that left/right translations are particularly useful when collision with doors occurs.

For each technique we determined the speed gain of the translation and rotation tasks through a pre-experiment with six subjects. We asked the participants to navigate inside our 3D virtual environment with each technique and to adjust the gains freely to feel comfortable when performing the task. We stopped the experiment and recorded the settings when the participant successfully went through 5 consecutive doors. Finally, for each technique, we averaged the values of gain between the participants. We noticed that the gain of the translation of the keyboard-mouse was higher than the 3D mouse or smartphone. This is probably due to the people habit in handling this technique.

C. Apparatus

The experiment was done in full-screen mode on a 24″ monitor with a resolution of 1920 by 1080 pixels. We developed the environment with a 3D open source engine, Irrlicht, in C++. For the keyboard and mouse device, we used a conventional optical mouse and a standard keyboard with 108 keys (Figure 16-a). For the 3D mouse we used the Space-Navigator [36] (Figure 16-b), a commercial device with 6 DOF. For the smartphone, we implemented the technique on a Samsung Galaxy S2 running Android 4.1.2 (Figure 16-c). To avoid an overload of the smartphone computing capacities with the processing of the internal sensors (accelerometers, gyroscope) we used an external 6D

tracker: the Polhemus Patriot Wireless [40] (Figure 16-d). We fixed a sensor to the rear face of the smartphone. Via a driver written in C++, the marker returns the position and the orientation of the smart-phone. We filtered the data noise with the 1€ filter [41].

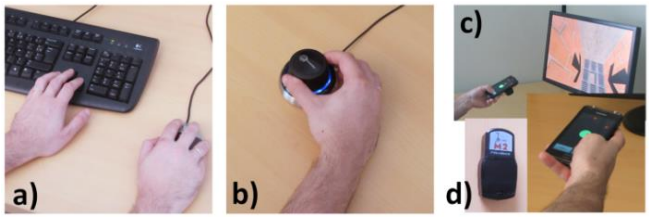


Figure 16. (a) The keyboard-mouse, (b) The 3D mouse and (c) the smartphone configuration. (d) The Polhemus sensor.

D. Participants and procedure

We recruited a group of 24 subjects (6 female) aged 29.3 (SD=9) on average. All subjects were used to the keyboard and mouse, 17 of them had a smartphone and only 1 had already used the 3D mouse.

Every participant performed the 3 techniques (smartphone, keyboard-mouse and 3D mouse). They started with the keyboard-mouse technique in order to be used as a reference. The order of smartphone and 3D mouse techniques was counterbalanced to limit the effect of learning, fatigue and concentration. For each technique, the subject navigated inside 6 different itineraries. We counterbalanced the itineraries associated with each technique across participants so that each technique was used repeatedly with each group of users.

The participants were siting during the experiment and were instructed to optimize the path, i.e., the distance travelled. They could train on each technique through one itinerary. When the user passed through a door, a positive beep was played. When the user collided with an edge of the tunnel, a negative beep was played.

After having completed the six trials for one technique, the subject filled the System Usability Scale (SUS) [42] and AttrakDiff [43] questionnaires and indicated three positive and negative aspects of the technique. The procedure is repeated for the two remaining technique. The experiment ended with a short interview to collect oral feedback. The overall duration of the experiment was about 1 hour and 30 minutes per participant.

E. Collected data

In addition to the SUS and AttrakDiff questionnaires filled after each technique to measure usability and attractiveness, we also asked for a ranking of the three interaction techniques in terms of preferences. From a quantitative point of view we measured the traveled distance and the number of collisions.

VI. RESULTS

In the following section here are the quantitative and qualitative results we obtained.

A. Quantitative results

First, a Kruskal-Wallis test confirmed that none of the 18 randomly chosen itineraries had an influence on the collected results. On average, we observed (Figure 17) that the travelled distance is the smallest with the keyboard-mouse (2766px, SD = 79), followed by the 3D mouse (2881px, SD = 125) and the smartphone (2996px, SD = 225). According to a Wilcoxon test these differences are significant. The same conclusions can be drawn with regard to the amount of collisions (keyboard-mouse: 5.08, SD = 5.68; 3D mouse: 16.11, SD = 15.86; smartphone: 33.35, SD = 24.64).

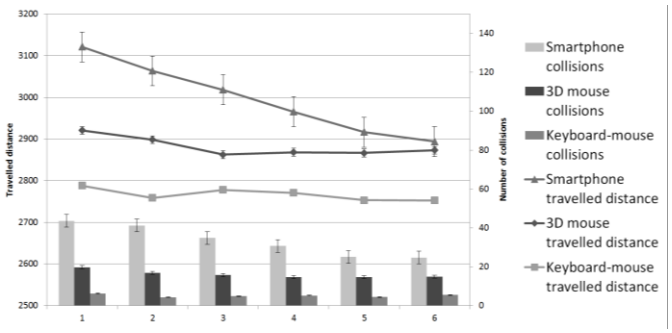


Figure 17. Evolution of the travelled distance and the amount of collisions according to 6 trials of the subjects.

Given the high dispersion of the distance and collision measures, we refined this analysis in distinguishing the results obtained for each of the six trials performed by the 24 participants (Figure 17). This refined analysis reveals a significant learning effect with the smartphone technique: between the first and sixth trial, the distance is 7.3% shorter (Wilcoxon test, $p = 6 \times 10^{-6}$) and collision are reduced of 43.3% (Wilcoxon test, $p = 2 \times 10^{-4}$). A significant learning effect is also observed with the 3D mouse, but only in terms of distance and with a smaller improvement (1.6% shorter, Wilcoxon test, $p = 0.049$).

The learning effect with the smartphone is so important that, at the last trial, the travelled distance for the smartphone (2893*px*) and the 3D mouse (2873*px*) is comparable (no significant difference, Wilcoxon test, $p = 0.49$).

B. Qualitative results

Three aspects have been considered in the qualitative evaluation: usability, attractiveness and the user's preference.

Usability evaluation: the SUS questionnaire [42] gives an average score of 82.60 (SD=12.90) for the keyboard-mouse, 54.79 (SD=22.47) for the smartphone based interaction and 53.54 (SD=27.97) for the 3D mouse. A Wilcoxon test shows that the SUS difference is statistically significant between the keyboard-mouse and each of the two other techniques (3D mouse, smartphone). However, the SUS difference is not statically significant between the 3D mouse and the smartphone. Research conducted to the interpretation of the SUS score [44] permits to classify the usability of the keyboard-mouse as "excellent". According to this interpretation scale, the usability of the smartphone and the 3D mouse is identified as "ok".

We also note a wide dispersion of the SUS score. We thus performed a more detailed analysis of the SUS score. First, according to [44] a system with a "good" usability must obtain a score above 70. In our experiment, 33% of the participants scored the 3D mouse above 70 while 37% of the participants scored the smartphone above 70. Second, 3D mouse and smartphone were two techniques unfamiliar to the participants. The results of the SUS questionnaire show that when the smartphone is used after the 3D mouse, the average score for the smartphone is 65.62 whereas in the other order the average score is 43.96. The perceived usability of the two unfamiliar techniques is therefore lower than the perceived usability of the keyboard-mouse. However, once the participants have manipulated these two unfamiliar techniques, the perceived usability of the smartphone increases drastically.

Attractiveness: Data collected using AttrakDiff [45] give an idea of the attractiveness of the technique and how it is experienced. Attrakdiff supports the evaluation of a system according to four distinct dimensions: the pragmatic quality (PQ: product usability, indicates if the users could achieve their goals using it); the hedonic quality – stimulation (HQ-S: determine to which extent the product can support the need in terms of new, interesting and stimulating functions, contents and interaction); the hedonic quality – identity (HQ-I: indicates to what extend the product allows the user to identity with it); the attractiveness (ATT: global values of the product based on the quality perception).

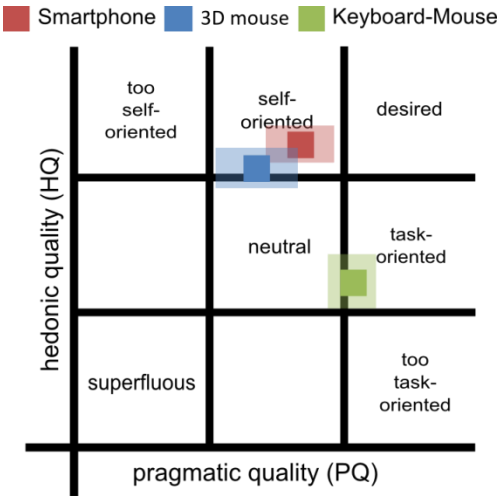


Figure 18. Portfolio generated on the AttrakDiff website.

Figure 18 shows a portfolio of the average value of the PQ and the HQ (HQ-S+HQ-I) for the three interaction techniques assessed in our user experiment.

The keyboard-mouse was rated as "fairly practice-oriented", i.e., one of the first levels in the "task-oriented" category. According to the website report [45], the average value of PQ (above 1) indicates that there is *definite* room of improvement in terms of usability. The average value of HQ obtained (approx. -1) expresses that there is *clearly* room for improvement in terms of user's stimulation. The 3D mouse was rated as "fairly self-oriented", i.e., one of the first levels in the category "self- oriented". The average value of PQ (approx. 0) expresses that there is room for improvement in terms of usability. The average value of HQ obtained (approx. 1) expresses that room for improvement also *exists* in terms of user's stimulation. The smartphone was rated as "self-oriented". The average value of PQ (approx. 0) expresses that there is room for improvement in terms of usability. The average value of HQ obtained (above 1) expresses that the user identifies with the product and is motivated and stimulated by it.

Figure 19 summarizes the average values for the four AttrakDiff dimensions of the three interaction techniques. With regards to the four dimensions the smartphone is rated higher than the 3D mouse and the differences are statistically significant (T-test, $p<0.05$). For the PQ value the keyboard-mouse is better than the smartphone (statistically significant, $p<0.05$). For HQ-I and HQ-S values the smartphone is better than the keyboard-mouse (statistically significant, $p<0.05$). In terms of ATT, the smartphone is again rated higher than keyboard-mouse but the difference is however not statistically significant ($p>0.05$). Compared to the keyboard-mouse, the smartphone is considered as novel, innovative, inventive, stylish and creative. Improvements in terms of simplicity, straightforwardness or predictability could increase the average value of PQ and probably increase even more the ATT value of the smartphone.

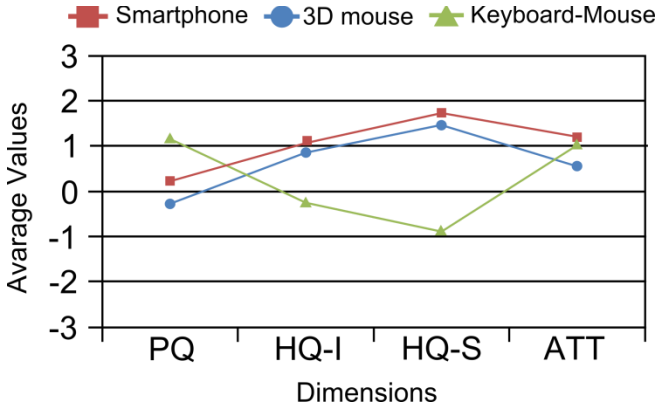


Figure 19. Average values for the four dimensions of the AttrakDiff questionnaire.

User preference: at the end of the experiment a short semi-guided interview was performed. The participants were first asked to rank the three techniques from 1 (best) to 3 (worst). The results are in line with the SUS scores: the keyboard-mouse technique is largely preferred and the 3D mouse is by far the least appreciated technique: only 2 participants out of 24 ranked it as the best, and 14 ranked it as the worst. The smartphone based-interaction is ranked uniformly in the three places (7, 9, 8).

Finally, three positive points and three negative points were asked for each technique. The most frequently

mentioned positive points are "quick", "easy" and "accurate" for the keyboard-mouse technique, "intuitive", "novel" and "usable with on hand" for the 3D mouse and "immersive", "funny", and "accessible to everybody" for the smartphone. The participants thus appreciate the conditions of use created by the smartphone while they particularly pinpoint the effectiveness of the mouse and provide general comments about the 3D mouse.

The most frequently mentioned negative aspect is related to a practical aspect of the keyboard-mouse ("requires the use of both hands"). They are related to the effectiveness of use of the 3D mouse ("difficulty to combine translation and rotation at the same time", "lack of precision" and "high need for concentration") and for the smartphone it focuses on one specific feature ("difficulty to translate to the left or right") and the overall context of use ("the apparent time of learning" and "the tiredness caused in the arm").

Technical issues for the 3D mouse and effectiveness of the keyboard-mouse are thus highlighted while the benefits and limits related to the interactive experience are mentioned for the smartphone. This clear shift of interest between the three techniques reveals that the disappointing performances of the smartphone highlighted in the previous section are not totally overruling the interest of the participants for the smartphone-based technique. It is therefore a very interesting proof of interest for further exploring the use of a smartphone in 3DVE.

VII. Discussion

A. Smartphone-based interaction: comparing our solution to existing ones

Among the existing attempts for exploring the navigation of 3DVE with a smartphone, two different settings exist. A first set of solutions, as opposed to our setting, proposes to display the 3DVE directly on the smartphone. Different techniques are explored to change the point of view inside the 3D scene: tactile screen like Navidget [46], integrated sensor [47], smartphone motions in the space around a reference, as Chameleon technique [48] and T(ether) [49] and manipulation of physical objects around the smartphone [50]. The second set of solutions avoids issues related to the occlusion of the 3DVE with fingers by displaying the 3DVE on a distant screen. These involved solutions integrated sensor [51] to detect user's motions, additional tactile screen [52] or a combination of both [53]. Although our technique is clearly in line with this second set of solutions, our use of the smartphone presents three major originalities. Firstly, the smartphone is not limited to a remote controller: it is also used to provide the user with feedback or personalized information. Secondly, using tactile interaction to support the navigation would occlude part of the screen and prevent its use to visualize data, selecting objects or clicking on additional features. Instead, physical gesture are applied to the smartphone to control rotations like in [54] or [53] but also to control translations of the point of view in the 3DVE. Thirdly, the choice of the gestures to apply has been guided by the results of a guessability study that highlights the most probable gesture users would perform with a smartphone. We used this approach rather than a pre-experiment or results of existing experiment [53] because when getting familiar with the manipulation of a smartphone, universal gestures will be adopted, and not necessarily the ones known as the most efficient. The users' prime intuition of use looked more important to us.

B. User's experiment results: analysis

Beyond the designed interaction technique, the contribution includes a set of evaluation results. The user experiment revealed a significant learning effect with the smartphone. This is a very encouraging result because no learning effect was observed with keyboard-mouse and 3D mouse although the participants were unfamiliar with 3D mouse and smartphone: the use of a smartphone thus significantly improves over the time.

Results also revealed that the use of our smartphone based technique to navigate inside a 3DVE is more attractive and stimulating than a more usual technique such as the keyboard-mouse and the 3D mouse.

In terms of usability, users' preferences (interaction technique ranking) and quantitatively (travelled distance and amount of collisions) our smartphone technique appears to be weaker than the keyboard-mouse technique but similar to the 3D mouse.

This tradeoff between attractivity and usability /performance emphasizes that compared to two manufactured devices; our technique is better accepted but weaker in performance and usability. This is particularly encouraging because technological improvements of our technique, such as mixing the use of integrated sensor with image processing to compute more robust and accurate smartphone position and orientation, will also increase the user's performance. In addition, the use of smartphones is already widely spread and we believe that their use as an interaction support with remote application will develop as well and become a usual interaction form.

Altogether this user experiment establishes that the use of a smartphone to interact with 3DVE is very promising and needs to be explored later.

VIII. Integrating the smartphone based technique and other advanced interaction techniques in the platform

A. Assessing the easy to plug feature of the platform

Following these very positive results, we have successfully and easily connected the presented smartphone interaction technique to the interactive platform presented in Section III. Concretely, our technique is used to control the navigation step in the pedagogical step of the scenario. It thus consists in moving through the dome and around the telescope to collect the inside temperature, displayed only when the visitor is close to the floor, and the outside temperature, displayed only when the visitor is close to the aperture of the dome. We just add one class implementing the *Controller* concept of the DLL to plug our interaction technique into the software platform. Approximately 10

minutes are sufficient to create the link between a new interaction technique and our software platform.

To further assess the ability of the platform to plug new interaction technique easily, we have developed two other advanced interaction techniques for controlling navigation. The first one is based on a physical doll (Figure 20 – left). An ARToolKit marker [55] and a push button are attached to the doll, which is handled by the user. When the user presses the button, the marker is tracked by a camera to trigger, in the interactive platform, a navigation command that corresponds to the doll motion. The user can thus move the doll to perform a translation in the 3D environment or a rotation of the point of view.

The second one is based on a physical cube (Figure 20 – right). Users' movements to navigate in the 3D environment are identical to the doll technique. We used the Polhemus Patriot Wireless [40] for tracking the position and the orientation of the cube over timer. We had a Phidget SBC board [56] for wirelessly transmitting the state of a push button. When the cube face presenting the rotation and translation instructions is facing upward, and if the button is pressed, any motion applied by the user to the physical cube will be directly mapped to a navigation command in the 3D environment. This design reveals the possibility to map other features to the five remaining cube faces. Again, inserting this different type of interaction technique, based on a new type of sensors, in the platform did not raise any problem of software connections.

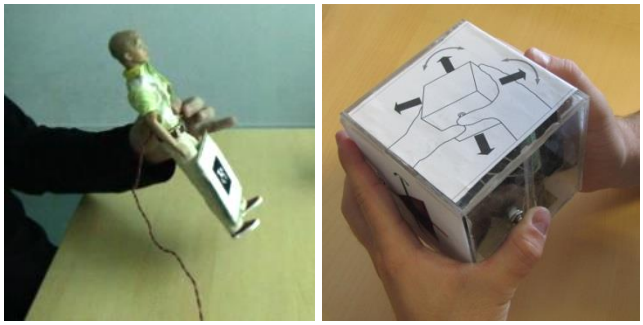


Figure 20. Tangible navigation technique based one a physical doll (left) and one a physical cube (right).

However, all the steps of the pedagogical scenario are not covered with these techniques. Indeed, the second and third steps that require selection and manipulation are not supported by these techniques.

Regarding the selection phase, as described in [57], we explored touchscreen input, mid-air movement of the mobile device (Figure 21-right), and mid-air movement of the hand around the device (Figure 21-left) for exploring and selecting element in 3D detail view of the virtual environment. Results shows that gesture with the smartphone or around the smartphone perform better than traditional tactile interaction modality. Interestingly, users preferred mid-air movement around the smartphone. Therefore, we developed an additional technique around the smartphone to support the selection step of the pedagogical scenario.

We also enriched the physical cube based interaction technique so that it covers the selection step by simply adding an RF-ID reader inside the cube and close to a second cube face. It thus supports the selection of a star to observe and thanks to the Phidget SBC board [56], the communication between the application and the RF-ID reader remains wireless.

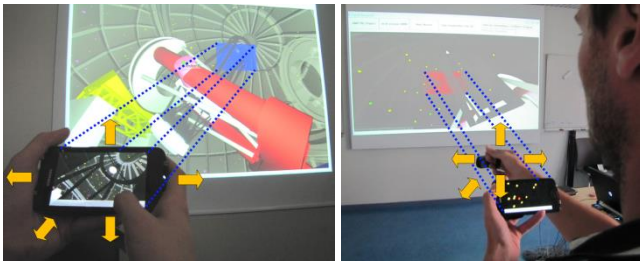


Figure 21. Smartphone-based technique for exploring and selecting element in 3D detail view: mid-air movement of the mobile device (left), and mid-air movement of the hand around the device (right).

Now regarding the manipulation step of the scenario, the only interaction techniques initially available were limited to the WIMP based controller described in Section III.C.2. We developed a smartphone application, which allows the user to manipulate and visualize the telescope information through a simple and easy to pick up interface (Figure 22). Sliders are displayed to allow the modification of the different parameters while a list of labels can be visualized to monitor the state of the relevant data.

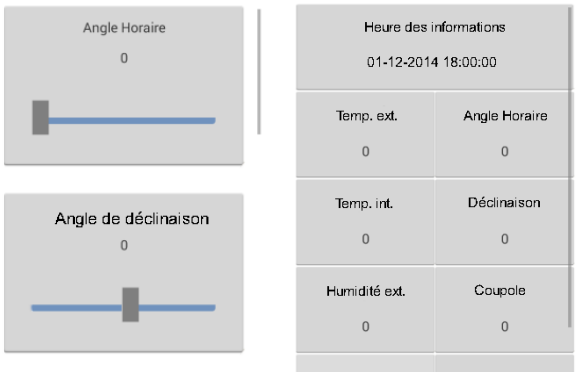


Figure 22. Smartphone interface for manipulation (left) and data visualization (right).

The cube based interaction techniques has also be enriched to support the manipulation task: two additional cube faces present the instructions to manipulate the different elements of the 3D environment. When facing upward, they respectively allow manipulating the dome elements and the telescope axes.

B. 3D Interactive Platform deployment

We deployed our interactive platform in two major public situations: in October 2013, during two days in our university hall for a scientific festival named Novela, and in June 2014, during two days in the museum of the Pic Du Midi (Figure 23). Large and varied audiences, ranging from

Figure 23. The in-situ deployment of our 3D interactive platform.

scholar to retired people, have used at this interactive installation. These two in-situ deployment permitted to validate the robustness of our platform and interaction techniques (WIMP, tangible and smartphone-based). It also allowed us to identify some limitations about the different techniques. Our future goal is to perform an overall user's experimentation of the different interaction techniques we plugged into the platform in order to assess and compare their learnability, efficiency with regards to the museum visit, usability and attractiveness.

IX. CONCLUSION

In this article, we have presented a software platform that provides a way to visualize, manipulate and navigate through a complex and concrete 3D environment representing the Telescope Bernard-Lyot. From the museum point of view, rather than using a poster or photographs, the interactive software platform provides an immersive and engaging way to transfer knowledge about the telescope to the museum visitors. The adapted MVC architecture grants a multiuser, multiplatform and remote access to the 3D environment, which is particularly interesting in a museum context. From the scientific point of view, the interactive software platform that we developed provides an easy way to plug new interaction techniques. In future works, other interaction modalities like tangible or gesture interaction could be plugged into the 3D in the 3D interactive platform. These techniques can then be tested and evaluated in a concrete 3D environment with real tasks (navigation, manipulation and visualization). The originality of this 3D museum exhibit lies in the fact that it has been designed to be opened and easy to adapt. As a result, our contribution to the field of interaction in 3D spaces is also a reference platform in which user-centered evaluation of other interaction techniques can be easily performed.

We then explored the feasibility of using a smartphone to navigate inside a 3DVE. Smartphones present the advantage to provide a private space for viewing and to constitute a personal device for navigating or controlling a 3D cultural or pedagogical content. Generalizing its use throughout a museum is also completely imaginable. With a QR code the visitor can easily download the mobile app in front of the exhibit and interact with the 3DVE. The originality of our technique relies on the fact that the smartphone is used as a tangible object. Physical actions on the smartphone trigger translation and rotation in the 3DVE. Very promising results have been highlighted in a user experiment comparing our solution to a keyboard-mouse technique and a 3D mouse, the most common devices found in museums nowadays. We measured that after a short learning period, the smartphone technique leads to performant results that are comparable with the 3D mouse. Through technical optimization we are also convinced that it might become comparable to the keyboard-mouse technique. But more notably, we clearly established that visitors find such a more attractive and stimulating solution. In this study, we have therefore established that the use of a smartphone as a tangible object for navigating inside a 3DVE is a good alternative to the keyboard-mouse and 3D mouse.

We then successfully plugged in our software platform different advanced interaction techniques that have then been used in public contexts. These techniques used tangible, tactile and gesture modality. To complete these works, it will be interesting to measure the impact of these different interaction techniques on a museum visit and on the quality of the educational transfer. For example, in terms of knowledge acquired through the defined pedagogical scenario, it could be original to compare smartphone-based technique with tangible interaction techniques.

In long term future work, we have two precise goals. First, we plan to add an interaction technique database within our software platform in which a description of every interaction technique evaluated would be stored with its associated results. This could be really beneficial for the future interaction techniques designers who will be able to compare their own technique with the already developed ones. The second objective is to reuse the platform for the monitoring of the energy consumption at the level of a university campus. The 3D environment will then be replaced by the simulation of the energy consumption in the campus. The visualization task would then provide a way to see the energy consumption and the manipulation task would allow the user to close windows or change the wall material for example. The software infrastructure would remain unchanged.

ACKNOWLEDGMENT

We would like to thank the Novela festival for supporting us in the deployment of the platform in a public space. We also thank the TBL team for their help and their implication in this project.

REFERENCE

[1] L.-P. Bergé, G. Perelman, M. Raynal, C. Sanza, M. Serrano, M. Houry-Panchetti, R. Cabanac, and E. Dubois, "Smartphone-based 3D navigation technique for use in a museum exhibit," in *The Seventh International Conference on Advances in Computer-Human Interactions (ACHI 2014)*, 2014, pp. 252–257.

[2] D. S. Tan, D. Gergle, P. Scupelli, and R. Pausch, "Physically large displays improve performance on spatial tasks," *ACM Trans. Comput. Interact.*, vol. 13, no. 1, pp. 71–99, Mar. 2006.

[3] "3D Ancient Wonders, archeological reconstruction online virtual museum." [Online]. Available: http://www.3dancientwonders.com/. [Accessed: 05-Feb-2014].

[4] R. Hawkey, "Learning with digital technologies in museums, science centres and galleries," *NESTA Futur.*, pp. 1–44, 2004.

[5] H. S. Jerome Rodrigues, "Transfer of spatial knowledge from virtual to real environment: effect of active/passive learning depending on a test-retest procedure and the type of retrieval tests," *J. Cybertherapy Rehabil.*, vol. 3, no. 3, pp. 275 – 285, 2010.

[6] L. Pecchioli, M. Carrozzino, F. Mohamed, M. Bergamasco, and T. H. Kolbe, "ISEE: Information access through the navigation of a 3D interactive environment," *J. Cult. Herit.*, vol. 12, no. 3, pp. 287–294, Jul. 2011.

[7] T. Wischgoll and J. Meyer, "An explorational exhibit of a pig's heart," in *ACM SIGGRAPH 2005 Posters on - SIGGRAPH '05*, 2005, p. 138.

[8] C. Christou, C. Angus, C. Loscos, A. Dettori, and M. Roussou, "A versatile large-scale multimodal VR system for cultural heritage visualization," in *Proceedings of the ACM symposium on Virtual reality software and technology - VRST '06*, 2006, pp. 133–140.

[9] H. Graf and K. Jung, "The smartphone as a 3D input device," in *2012 IEEE Second International Conference on Consumer Electronics - Berlin (ICCE-Berlin)*, 2012, pp. 254–257.

[10] H.-N. Liang, J. Trenchard, M. Semegen, and P. Irani, "An exploration of interaction styles in mobile devices for navigating 3d environments," in *Proceedings of the 10th asia pacific conference on Computer human interaction - APCHI '12*, 2012, pp. 309–313.

[11] D. Avrahami, J. O. Wobbrock, and S. Izadi, "Portico: tangible interaction on and around a tablet," in *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*, 2011, pp. 347–356.

[12] O. Shaer and E. Hornecker, "Tangible User Interfaces: past, present, and future directions," *Found. Trends® Human–Computer Interact.*, vol. 3, no. 1–2, pp. 1–137, Jan. 2009.

[13] A. Parush and D. Berman, "Navigation and orientation in 3D user interfaces: the impact of navigation aids and landmarks," *Int. J. Hum. Comput. Stud.*, vol. 61, no. 3, pp. 375–395, Sep. 2004.

[14] C.-Y. Lin, "Investigating the potential of on-line 3D virtual environments to improve access to museums as both an informational and educational resource," Faculty of Art and Design De Montfort University, 2009.

[15] L. Chittaro, L. Ieronutti, and R. Ranon, "Navigating 3D virtual environments by following embodied agents: a proposal and its informal evaluation on a virtual museum application," *PsychNology J.*, vol. 2, no. Special issue on Human-Computer Interaction, pp. 24–42, 2004.

[16] P. Alessio and A. Topol, "A public 3D visualization tool for the musée des Arts et Métiers de Paris," in *ICEC'11 Proceedings of the 10th international conference on Entertainment Computing*, 2011, pp. 136–142.

[17] P. Mulholland, T. Collins, and Z. Zdrahal, "Bletchley Park Text: Using mobile and semantic web technologies to support the post-visit use of online museum resources," *J. Interact. Media Educ.*, no. Special Issue: Portable Learning - Experiences with Mobile Devices, pp. 1–21, 2005.

[18] T. Barbieri, F. Garzotto, G. Beltrame, L. Ceresoli, M. Gritti, and D. Misani, "From dust to stardust: a collaborative 3D virtual museum of computer science," in *International Cultural Heritage Informatics Meeting - ICHIM*, 2001, pp. 341–345.

[19] G. Jansson, M. Bergamasco, and A. Frisoli, "A new option for the visually impaired to experience 3D art at museums: manual exploration of virtual copies," *Vis. Impair. Res.*, vol. 5, pp. 1–12, 2003.

[20] R. Berndt, D. W. Fellner, and S. Havemann, "Generative 3d models: a key to more information within less bandwidth at higher quality," in *Web3D '05*, 2005, pp. 111–121.

[21] J. Taylor, J.-A. Beraldin, G. Godin, L. Cournoyer, R. Baribeau, F. Blais, M. Rioux, and J. Domey, "NRC's 3D imaging technology for museum & heritage applications," *J. Vis. Comput. Animat.*, vol. 14, pp. 121–138, 2006.

[22] C. H. Esteban and F. Schmitt, "Silhouette and stereo fusion for a 3d object modelling," *Comput. Vis. Image Underst.*, vol. 96, no. 3, pp. 367–392, 2003.

[23] M. Glencross, A. G. Chalmers, M. C. Lin, M. A. Otaduy, and D. Gutierrez, "Exploiting perception in high-fidelity virtual environments," in *ACM SIGGRAPH 2006 Courses on - SIGGRAPH '06*, 2006, pp. 1–188.

[24] J.-A. Beraldin, M. Picard, S. F. El-Hakim, G. Godin, V. Valzano, and A. Bandiera, "Combining 3D technologies for cultural heritage interpretation and entertainment," in *SPIE 5665, Videometrics VIII*, 2005, pp. 108–118.

[25] E. Ciabatti, P. Cignoni, C. Montani, and R. Scopigno, "Towards a distributed 3D virtual museum," in *Proceedings of the working conference on Advanced visual interfaces - AVI '98*, 1998, pp. 264–266.

[26] X. Guo and P. W. H. Chung, "The architecture of a web service-based remote control service system," in *Proceedings of the 10th International Conference on Information Integration and Web-based Applications & Services - iiWAS '08*, 2008, pp. 555–558.

[27] M. White, N. Mourkoussis, J. Darcy, P. Petridis, F. Liarokapis, P. Lister, K. Walczak, R. Wojciechowski, W. Cellary, J. Chmielewski, M. Stawniak, W. Wiza, M. Patel, J. Stevenson, J. Manley, F. Giorgini, P. Sayd, and F. Gaspard, "ARCO - an architecture for digitization, management and presentation of virtual exhibitions," in *Proceedings Computer Graphics International, 2004.*, 2004, pp. 622–625.

[28] M. D. Dickey, "Brave new (interactive) worlds: A review of the design affordances and constraints of two 3D virtual worlds as interactive learning environments," *Interact. Learn. Environ.*, vol. 13, no. 1–2, pp. 121–137, Apr. 2005.

[29] F. Liarokapis, S. Sylaiou, A. Basu, N. Mourkoussis, M. White, and P. F. Lister, "An interactive visualisation interface for virtual museums," *Eurographics Assoc. p. 47-56*, pp. 47–56, Dec. 2004.

[30] C.-W. Fu, W.-B. Goh, and J. A. Ng, "Multi-touch techniques for exploring large-scale 3D astrophysical simulations," in *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*, 2010, pp. 2213–2222.

[31] E. Woods, M. Billinghurst, J. Looser, G. Aldridge, D. Brown, B. Garrie, and C. Nelles, "Augmenting the science centre and museum experience," in *Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Austalasia and Southe East Asia - GRAPHITE '04*, 2004, pp. 230–236.

[32] M. Baron, V. Lucquiaud, D. Autard, and D. L. Scapin, "K-MADe: un environnement pour le noyau du modèle de description de l'activité," in *Proceedings of the 18th international conference on Association Francophone d'Interaction Homme-Machine - IHM '06*, 2006, pp. 287–288.

[33] "Home | SketchUp," 2014. [Online]. Available: http://www.sketchup.com/. [Accessed: 03-Jul-2014].

[34] "Irrlicht Engine - A free open source 3D engine," 2014. [Online]. Available: http://irrlicht.sourceforge.net/. [Accessed: 14-Feb-2014].

[35] "The Ivy software bus." [Online]. Available: http://www.eei.cena.fr/products/ivy/. [Accessed: 16-Feb-2014].

[36] "3Dconnexion : SpaceNavigator." [Online]. Available: http://www.3dconnexion.fr/products/spacenavigator. [Accessed: 27-Jan-2014].

[37] J. O. Wobbrock, M. R. Morris, and A. D. Wilson, "User-defined gestures for surface computing," in *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*, 2009, p. 1083.

[38] T. Tsandilas, E. Dubois, and M. Raynal, "Modeless pointing with low-precision wrist movements," in *Human-Computer Interaction – INTERACT 2013*, 2013, vol. 8119, pp. 494–511.

[39] A. Martinet, G. Casiez, and L. Grisoni, "The effect of DOF separation in 3D manipulation tasks with multi-touch displays," in *Proceedings of the 17th ACM Symposium on Virtual Reality Software and Technology - VRST '10*, 2010, p. 111.

[40] "Polhemus Patriot Wireless - Polhemus," 2014. [Online]. Available: http://www.polhemus.com/motion-tracking/all-trackers/patriot-wireless/. [Accessed: 14-Feb-2014].

[41] G. Casiez, N. Roussel, and D. Vogel, "1 € filter: a simple speed-based low-pass filter for noisy input in interactive systems," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*, 2012, pp. 2527–2530.

[42] J. Brooke, "SUS: A quick and dirty usability scale," *Usability Eval. Ind.*, pp. 189–194, 1996.

[43] M. Hassenzahl, "The interplay of beauty, goodness, and usability in interactive products," *Human-Computer Interact.*, vol. 19, no. 4, pp. 319–349, Dec. 2004.

[44] A. Bangor, P. T. Kortum, and J. T. Miller, "An empirical evaluation of the System Usability Scale," *Int. J. Hum. Comput. Interact.*, vol. 24, no. 6, pp. 574–594, Jul. 2008.

[45] "AttrakDiff." [Online]. Available: http://attrakdiff.de/index-en.html. [Accessed: 27-Jan-2014].

[46] M. Hachet, F. Decle, S. Knodel, and P. Guitton, "Navidget for 3D interaction: Camera positioning and further uses," *Int. J. Hum. Comput. Stud.*, vol. 67, no. 3, pp. 225–236, 2009.

[47] W. Hürst and M. Helder, "Mobile 3D graphics and virtual reality interaction," in *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology - ACE '11*, 2011, pp. 28–36.

[48] B. Buxton and G. W. Fitzmaurice, "HMDs, caves & chameleon: A human-centric analysis of interaction in virtual space," *Comput. Graph. SIGGRAPH Q.*, vol. 32, no. 4, pp. 64–68, 1998.

[49] D. Lakatos, M. Blackshaw, A. Olwal, Z. Barryte, K. Perlin, and H. Ishii, "T(ether): spatially-aware handhelds, gestures and proprioception for multi-user 3D modeling and animation," in *Proceedings of the 2nd ACM symposium on Spatial user interaction - SUI '14*, 2014, pp. 90–93.

[50] M. Hachet, J. Pouderoux, and P. Guitton, "3D elastic control for mobile devices," *IEEE Comput. Graph. Appl.*, vol. 28, no. 4, pp. 58–62, Jul. 2008.

[51] S. Boring, M. Jurmu, and A. Butz, "Scroll, tilt or move it: using mobile phones to continuously control pointers on large public displays," in *Proceedings of the 21st Annual Conference of the Australian Computer-Human Interaction Special Interest Group on Design: Open 24/7 - OZCHI '09*, 2009, pp. 161–168.

[52] D. Gracanin, K. Matkovic, and F. Quek, "iPhone/iPod Touch as Input Devices for Navigation in Immersive Virtual Environments," in *2009 IEEE Virtual Reality Conference*, 2009, pp. 261–262.

[53] A. Benzina, A. Dey, M. Toennis, and G. Klinker, "Empirical evaluation of mapping functions for navigation in virtual reality using phones with integrated sensors," in *Proceedings of the 10th asia pacific conference on Computer human interaction - APCHI '12*, 2012, pp. 149–158.

[54] F. Daiber, L. Li, and A. Krüger, "Designing gestures for mobile 3D gaming," in *Proceedings of the 11th International Conference on Mobile and Ubiquitous Multimedia - MUM '12*, 2012, pp. 3–8.

[55] P. Lamb, "ARToolKit developer homepage." [Online]. Available: http://artoolkit.sourceforge.net/. [Accessed: 29-Jul-2014].

[56] Phidgets, "Phidgets Inc. - 1073_0 - PhidgetSBC3." [Online]. Available: http://www.phidgets.com/products.php?category=21&product_id=1073_0. [Accessed: 29-Jul-2014].

[57] L.-P. Bergé, M. Serrano, G. Perelman, and E. Dubois, "Exploring smartphone-based interaction with overview+detail interfaces on 3D public displays," in *Proceedings of the 16th international conference on Human-computer interaction with mobile devices & services - MobileHCI '14*, 2014, pp. 125–134.

A Hybrid Ant Colony and Branch-and-Cut Algorithm to Solve the Container Stacking Problem at Seaport Terminal

Ndèye Fatma Ndiaye
University of Le Havre
Laboratory of applied mathematics
25 Rue Philippe Lebon
76600 Le Havre cedex, France
Email: farlou@live.fr

Adnan Yassine
Superior institute of logistics studies
Quai Frissard, B.P. 1137
76063 Le Havre cedex, France
Email: adnan.yassine@univ-lehavre.fr

Ibrahima Diarrassouba
University Institute of Technology
Place Robert Schuman
76600 Le Havre cedex
France
Email: diarrasi@univ-lehavre.fr

***Abstract*—The container storage problem is one of the most studied issues regarding seaports. It is a relevant problem due to the fact that the effectiveness of a storage yard management affects the global productivity of the port. Therefore, various attempts were done in order to elaborate efficient decision support systems, which concern specific container terminals and specific transfer and handing equipments. Most of the existing proposed methods use heuristic or meta-heuristic algorithms because the NP-hardness of the container storage problem makes it difficult to solve using exact optimization methods mainly when there are a lot of containers. In this paper, we combine an exact resolution method (branch-and-cut) and a meta-heuristic algorithm (ant colony) in a hybrid ant colony and branch-and-cut algorithm (HACBC). Numerical simulations prove the efficiency and the effectiveness of our algorithm.**

Keywords–Ant colony algorithm; Branch-and-cut; Container storage problem; Hybridization; Mathematical modelling.

I. INTRODUCTION

In a seaport, the container terminal manages all actions concerning containers. Generally, three types of containers are distinguished: outbound, inbound, and transshipment containers. All these containers are temporarily stacked in the container yard, before leaving the port. Outbound containers are brought by External Trucks (ETs), also picked up by the Straddle Carriers (SCs), which store them in their storage locations, and then loaded onto vessels. Inbound containers are unloaded from vessels by the Quai Cranes (QCs), transported to their storage locations by the SCs, and then recuperated later by ETs. Transshipment containers come to the port by ship and also leave the port by ship, after spending their dwell times in the storage yard.

Nowadays, the competition between ports is very high. Therefore, each of them tries to improve continuously the quality of its service in order to attract more customers. The most important criteria to measure service level, include the waiting time of ETs, which collect inbound containers. In fact, when an ET arrives at port and claims a specific container, it waits during all the time required to retrieve it. If the desired container is under others, it may be necessary to move firstly these containers. This kind of movements, named reshuffles, are unproductive and time consuming. Therefore, it is very important to optimally store containers. Another important criterion to measure the quality of service is the time required to unload ships. The importance of this factor is justified by the fact that it is more beneficial for both the port and the customers to shorten the stay of vessels. On one hand, it is better for the port authorities to quickly free the berths in order to allocate them to others incoming vessels. On other hand, generally shipowners rent vessels. Therefore, they tend to minimize the berthing durations in order to increase their profits. These two issues are addressed in this paper.

We consider a modern container terminal, which uses SCs instead of Internal Trucks (ITs). The advantage of a SC is the fact that it is able to lift and to store a container itself. Therefore, it is not necessary to use Yard Cranes (YCs). A storage yard is composed of several blocks. In order to enable the circulation of the SCs, each block is made up of several bays, which are separated by small spaces. In every bay, there are stacks wherein containers are stored. A stack must have a height inferior or equal to the limit fixed by the port authorities. Figure 1 shows an example of block wherein circulate straddle carriers.

Figure 1. Straddle carriers circulating in a containers yard

In this paper, we tackle the storage of inbound containers in a seaport terminal. We propose an efficient storage method, which enables to store the containers without causing no reshuffle. A linear mathematical model, which determines an

accurate storage location for each container is designed for this purpose. This mathematical model minimizes the total distance travelled by the straddle carriers from the quays to the storage yard. A branch-and-cut algorithm (BC-CSP) was proposed in [1] for the resolution of this problem. In this paper, we improve this algorithm by combining it with an ant colony algorithm.

The remainder of the paper is organized as follows: a literature review is given in Section II, a detailed description of the addressed problem is exposed in Section III, the mathematical model is explained in Section IV, the complexity of the problem is discussed in Section V, the branch-and-cut algorithm is itemized in Section VI, the ant colony algorithm is explained in Section VII, the hybrid ant colony and branch-and-cut algorithm is detailed in Section VIII, the numerical results are presented in Section IX, a conclusion is given in Section X.

II. Literature review

There are more papers addressing the storage of outbound containers than inbound containers. However, there are some papers that deal with both simultaneously. In [2], Zhang et al. considered in addition to these two categories of container, those that are in transition, that means the containers that are unloaded from some vessels and are waiting for being loaded onto other ships. They used the rolling-horizon approach to solve the storage space allocation problem. For each planing horizon, they solved the problem in two steps that are formulated as mathematical programs. In the first step, they determined the total number of containers that must be assigned to each block at a period so that the workload of loading and unloading of each vessel are balanced. Then, in the second step they determined the number of containers that must be associated to every vessel in order to minimize the total distance travelled to transport these containers from the quays to the storage blocks. In [3], Bazzazi et al. proposed a genetic algorithm to solve an extended version of the storage space allocation problem (SSAP). It consisted to allocate temporarily locations to the inbound and outbound containers in the storage yard according to their types (regular, empty, and refrigerated). They aimed to balance the workloads of the blocks with the goal to minimize the time required to store or to retrieve containers. In [4], Park et al. dealt with the planar storage location assignment problem (PSLAP), in which only planar movements were allowed. The purpose of the PSLAP was to store inbound and outbound containers so as to minimize the number of moving obstructive objects. The authors made a mathematical formulation of the PSLAP and proposed a genetic algorithm to solve it. In [5], Lee et al. combined the truck scheduling and the storage allocation problems. They considered inbound and outbound containers, and attempted to minimize the weighted sum of the total delay of requests and the total travel time of the yard trucks. For the numerical resolution, they proposed a hybrid insertion algorithm. In [6], Kozan et al. developed an iterative search algorithm by using a transfer model and an assignment model. At first, the algorithm determined cyclically the optimum storage locations for inbound and outbound containers, and secondly it found the corresponding handling schedule. They solved the problem by a genetic algorithm, a tabu search algorithm and a hybrid algorithm.

Concerning inbound containers, most of the papers dealt with the management of reshuffles. In [7], Sauri et al. proposed three different strategies to store inbound containers. The purpose of their work was to determine the best strategy that minimizes re-handles in an import container yard. For this, they developed a mathematical model based on probabilistic distribution functions to evaluate the number of reshuffles. In [8], Kim et al. considered a segregation strategy to store inbound containers. This method did not allow to place newly arriving containers over those that arrived earlier. Therefore, storage spaces are allocated to each vessel in order to minimize the number of expected reshuffles during the loading operations. In [9], Cao et al. proposed an integer programming model, which addressed the trucks scheduling and the storage of inbound containers. They minimized simultaneously the number of congestions, the waiting time of trucks, and the unloading time of containers. The authors designed a genetic algorithm to solve the model, and another heuristic algorithm, which outperformed their genetic algorithm. In [10], Yu et al. treated the storage problem of inbound containers in a modern automatic container terminal. They aimed to minimize the number of reshuffles in two steps. For this, they firstly resolved the block space allocation problem for the newly arriving inbound containers, and then, after the retrieving of some containers, they tackled the re-marshalling processes in order to re-organize the block space allocation. They suggested three mathematical models of storage containers, the first was a non-segregation model, the second was a single-period segregation model, and the third was a multiple-period segregation model. They conceived a convex cost network flow algorithm for the first and the second models, and a dynamic programming for the third. They found out that the re-marshalling problem is NP-hard, and then, they designed a heuristic algorithm to solve it. In [11], Moussi et al. considered a container terminal wherein reshuffles are not allowed. They proposed a new mathematical model to allocate storage spaces to inbound containers in such a way that no reshuffle will be necessary to retrieve them later. They designed a hybrid algorithm including genetic algorithm and simulated annealing to solve it. Ndiaye et al. strengthened that work by proposing in [1] a branch-and-cut algorithm, which is an exact optimization method, unlike the hybrid genetic and simulated annealing algorithm.

In most container terminals, the departure time of an inbound container is generally unknown. Kim et al. considered in [12] a container terminal, in which there is a limited free time storage for inbound containers, beyond which customers have to pay storage costs. The authors proposed a mathematical model to find the optimal price schedule.

Papers that dealt with the storage problem of outbound containers have generally different goals. In [13], Preston et al. proposed a container location model (CLM) to store outbound containers in a manner that minimised the time service of container ships. They designed a genetic algorithm for the numerical resolution. In [14], Kim et al. developed a dynamic programming model to determine storage locations for outbound containers according to their weights. They minimized the number of relocations expected during the loading operations of ships. They also made a decision tree using the set of optimal solutions to support real-time decisions. In [15], Chen et al. addressed in two steps the storage space allocation problem of outbound containers. In

the first step, they used a mixed integer programming model to calculate the number of yard bays and the number of locations in each of them. So, in the second step, they determined, for each container, the exact location where it will be stored. In [16], Woo et al. proposed a method to allocate storage spaces to groups of outbound containers. They reserved, for each group of containers that have the same attributes, a collection of adjacent stacks. At the end, the authors proposed a method to determine the necessary amount of storage spaces expected for all the outbound containers. In [17], Kim et al. gave two linear mathematical models to store outbound containers. In the first, they considered a direct transfer system. And then, in the second, they dealt with an indirect transfer system. They designed two heuristic algorithms to solve these models. The one was based on the duration-of-stay of containers, while they used the sub-gradient optimization technique in the other.

One of the few papers that dealt only with transshipment containers is that of Nishimura et al. [18]. The authors developed an optimization model to store temporarily transshipment containers in the storage yard, and proposed a heuristic based on Lagrangian relaxation method for the numerical resolution.

To the best of our knowledge, there is no paper that combines a meta-heuristic algorithm and an exact optimization method for the resolution of the container storage problem. Since the exact methods are not able to solve quickly large instances due to the NP-hardness of the problem, hybridization with meta-heuristic is a way to speed up the computation processes. So, in this paper, we exploit this idea by proposing a hybridization concerning a branch-and-cut algorithm and an ant colony algorithm.

III. CONTEXT

When a container ship arrives at port, the QCs unload the inbound containers and then place them on quays. After that, they are picked up by the SCs, which carry and store them in the container yard. The containers are picked up following the same order that they are unloaded from the ships. In order to avoid congestion at quays, which could increase the time required to unload the ships, we minimize the total distance travelled by the SCs between the quays and the container yard. In this study, we consider the following five hypotheses:
(1) reshuffles are not allowed,
(2) in each stack, the containers are stored following:
(2.a) the same order that they are unloaded from the ships,
(2.b) and the descending order of their departure times,
(3) in a stack, the containers have similar dimensions,
(4) we take into account the containers that are already present in the storage yard at the beginning of the current storage period,
(5) we do not exceed the maximum capacity of each stack.

Notice that the unloading order of the containers from a ship is decided by the port authorities before the arrival of this later at port. The role of such unloading plan is to ensure the stability of the ship during the unloading operations. However, the determination of the unloading plan and the storage plan (of inbound containers in the yard) are done separately, even if the results of the first problem are used for the resolution of the second. In this paper, we focus on the determination of the optimal storage plan of inbound containers in the yard.

IV. MATHEMATICAL MODELLING

In this section, we present a mathematical model that allocates an accurate storage location to every container. For this, we use the following indices, parameters, and decision variables.

Indices

k : container.
p : stack.
i : location in a stack.

Parameters

N_p : total number of stacks in the terminal.
c_p : number of available locations in the stack p.
r_p : size of the stack p, (20-feet, 40-feet, 45-feet, etc.).
t_p : departure time of the container that is at the top of the stack p at the beginning of the current storage period. It is equal to M if the stack is empty.
N : total number of inbound containers at quays.
T_k : departure time of the container k.
R_k : size of the container k, (20-feet, 40-feet, 45-feet, etc.).
d_p^k : distance between the stack p and the quay where is the inbound container k.
O_k : unloading order of the container k from ships.
M : a great integer.

Decision variables

$$x_{p,i}^k = \begin{cases} 1 & \text{If the location } i \text{ of the stack } p \text{ is allocated to the container } k. \\ 0 & \text{Otherwise.} \end{cases}$$

Model

$$\min \sum_{k=1}^{N} \sum_{p=1}^{N_p} \sum_{i=1}^{c_p} d_p^k x_{p,i}^k \quad (1)$$

The objective function (1) minimises the total distance travelled by the straddle carriers between the quays and the storage yard.

$$\sum_{p=1}^{N_p} \sum_{i=1}^{c_p} x_{p,i}^k = 1, \ \forall\, k = 1, ..., N \quad (2)$$

Constraint (2) ensures that each container is assigned to a single storage location.

$$\sum_{k=1}^{N} x_{p,i}^k \leq 1, \ \forall\, p = 1, ..., N_p, \ i = 1, ..., c_p \quad (3)$$

Constraint (3) secures that several containers are not assigned simultaneously to a same storage location.

$$\sum_{p=1}^{N_p} \sum_{i=1}^{c_p} x_{p,i}^k = 0, \ \forall\, k = 1, ..., N \ : \ R_k \neq r_p \text{ or } T_k > t_p \quad (4)$$

Constraint (4) guarantees that a container can be assigned to a stack only if they are compatible. In other words, if the container and the stack have similar dimensions (20-feet, 40-feet, 45-feet, etc.), and if the departure time of the container is inferior or equal to that of the container that is already at the top of the stack (at the beginning of the new storage period).

$$\sum_{k=1}^{N}(N-O_k+1)x_{p,i}^{k} \geq \sum_{k=1}^{N}(N-O_k+1)x_{p,i+1}^{k} \quad (5)$$
$$\forall\, p=1,...,N_p,\ i=1,...,c_p-1$$

Constraint (5) has two roles. The first is to enforce that, in every stack, the containers are stored following the ascending order of their unloading numbers. This avoids congestion at quays. The second is to secure that each stack is filled from bottom to top without omitting no location. This enables to satisfy the gravity's law, and excludes unrealisable assignments like the one that is shown in the example bellow.

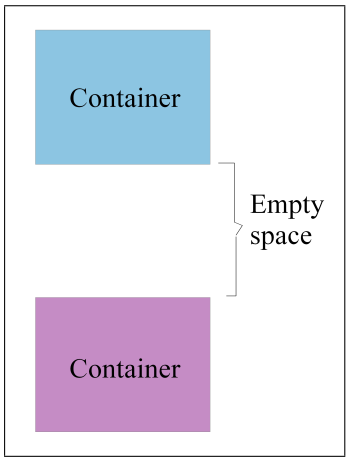


Figure 2. Unrealisable storage

$$\sum_{k=1}^{N}T_k x_{p,i}^{k} \geq \sum_{k=1}^{N}T_k x_{p,i+1}^{k} \quad (6)$$
$$\forall\, p=1,...,N_p;\ i=1,...,c_p-1$$

Constraint (6) ensures that, in every stack, the containers are stored following the descending order of their departure times. This avoids reshuffles when extracting containers.

$$x_{p,i}^{k} \in \{0,1\}, \quad \forall\, p=1,...,N_p$$
$$\forall\, i=1,...,c_p,\ k=1,...,N \quad (7)$$

Constraint (7) states that the decision variables are boolean.

V. Complexity of the problem

In this section, we study the complexity of the container storage problem (CSP). In particular, we show that it is equivalent to the bounded colouring problem (BCP). Therefore, it is NP-hard in the general case.

A. Some reminders about the BCP

Let us begin by recalling some concepts and definitions that will be useful for the following.

1) Preliminary notions: Let $G(V,E)$ be an undirected graph, V is the set of vertices and E is the set of edges.

G is a *comparability* graph if and only if there is a sequence of vertices $v_1,...,v_n$ of V such that for each (p,q,r) checking $1<p<q<r<n$, if $(v_p,v_q)\in E$ and $(v_q,v_r)\in E$, then $(v_p,v_r)\in E$.

A *co-comparability* graph is the complement of a comparability graph.

An undirected graph $G=(V,E)$ is a *permutation* graph if and only if there is a sequence of vertices $v_1,...,v_n$ of V and a permutation σ of the vertices such that: $\forall\, i,j\in\{1,...,n\}$, satisfying $1\leq i<j\leq n$, we have $(v_i,v_j)\in E$ if and only if $\sigma(i)>\sigma(j)$.

Theorem 1: A graph G is a permutation graph if and only if G and its complement are comparability graphs [19].

2) The bounded colouring problem: Given an undirected graph $G=(V,E)$, a set of s colours $l_1,...,l_s$, an integer H and a vector that gives the weight of assigning a colour l_i to a vertex of the graph. Solve the bounded colouring problem with minimum weight consists to determine a minimum weight colouring of G by using at most s colours in such a way that a colour is assigned to at most H vertices.

Theorem 2: The bounded colouring problem with minimum weight is NP-hard in the case of permutation graphs if $H\geq 6$ [20].

B. Case of storage where the stacks are empty at the beginning

In this section, we consider a case of storage where each stack of the storage yard is empty at the begin of the current storage period. We show that the CSP is NP-hard. For this, we introduce an undirected graph $G(N,O,T)=(V,E)$, which is constructed using an instance of the CSP, where N is the set of containers and O and T are two vectors that give respectively the unloading order and the departure time of each container. The graph G is constructed as follows. A vertex of the graph corresponds to a container. To simplify the notations, the index k is used to denote as well a container as the vertex that corresponds to it in the graph. There is an edge between two vertices k and k' if at least one of these two following conditions is satisfied:

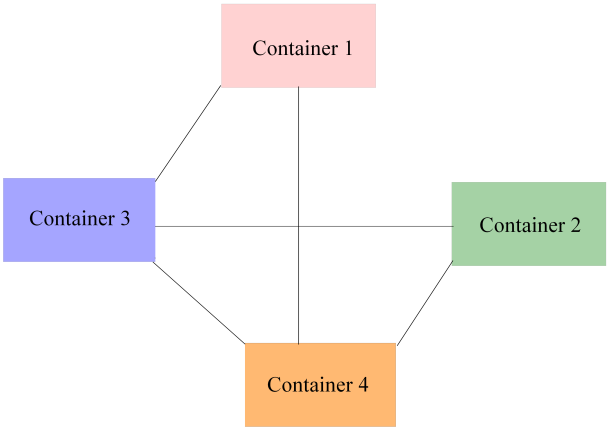
- $O_k<O_{k'}$ and $T_k<T_{k'}$,
- $R_k\neq r_p$.

Figure 3. is an example of graph constructed using an instance of the CSP.

We have the following lemma.

Lemma 1: In the case where the containers have similar dimensions, the graph $G(N,O,T)$ obtained from an instance of the CSP is a permutation graph.

Proof: To prove the fact that the graph $G(N,O,T)$ is a permutation graph, it suffices to show that it is a comparability graph as well as its complement (see Theorem 1).



	Dimension	Departure time	Unloading order
Container 1	20-feet	10	1
Container 2	20-feet	7	2
Container 3	20-feet	12	3
Container 4	40-feet	17	4

Figure 3. Graph constructed using an instance of the CSP.

Firstly, we show that $G(N, O, T)$ is a comparability graph. The vertices are ordered following the same order that the containers are unloaded from the ships. If two containers k and k' are unloaded from different ships at the same time (that is to say: $O_k = O_{k'}$), then the vertices k and k' are ordered in the ascending order of their departure times. If $O_k = O_{k'}$ and $T_k = T_{k'}$ then the vertices are ordered in the lexicographical order. Without loss of generality, we consider that the vertices are arranged in the order that is previously determined. Now, consider any three vertices k, k' and k'' of the graph, such that $k < k' < k''$, $(k, k') \in E$ and $(k', k'') \in E$. We will prove that necessarily $(k, k'') \in E$. As $(k, k') \in E$ and $(k', k'') \in E$, we have $O_k < O_{k'}$ and $T_k < T_{k'}$, and we have also $O_{k'} < O_{k''}$ and $T_{k'} < T_{k''}$. We thus obtain that $O_k < O_{k'} < O_{k''}$ and $T_k < T_{k'} < T_{k''}$, which implies that the graph $G(N, O, T)$ has an edge between the vertices k and k''. So $G(N, O, T)$ is a comparability graph.

At present, we will prove that the complement of $G(N, O, T)$, denoted $\bar{G}(N, O, T)$ is also a comparability graph. Firstly, notice that there is an edge between two vertices k and k' of $\bar{G}(N, O, T)$ if and only if there is no edge in $G(N, O, T)$ between k and k' (in other words $O_k < O_{k'}$ and $T_k > T_{k'}$). The vertices of $\bar{G}$ are ordered in the same order as those of G. As before, for any three vertices k, k', and k'' of the graph $\bar{G}(N, O, T)$, such that $k < k' < k''$, $(k, k') \in E$ and $(k', k'') \in E$, we have $O_k < O_{k'} < O_{k''}$ and $T_k > T_{k'} > T_{k''}$. So, $O_k < O_{k''}$ and $T_k > T_{k''}$, thus, there is an edge between k and k'' in $\bar{G}(N, O, T)$. Therefore, $\bar{G}(N, O, T)$ is also a comparability graph. ■

Now, it is easy to see that a solution of the container storage problem is a solution of the corresponding bounded colouring problem. In fact, a similar result is given in [21]. Consider an instance $ICSP = (N, O, T, N_p, H, r, R, d)$ of the CSP and the graph $G(N, O, T)$ associated to it, H is the maximum number of containers allowed in a stack. Consider a H-colouring of $G(N, O, T)$, which has s colours. Each colour of the bounded colouring problem is matched to a stack of the CSP. Indeed, as all vertices that have the same colour form a stable set, in other words they are not connected by any edges. Therefore, any two containers k and k' corresponding to two vertices of this stable set can be stored in a same stack because they satisfy these two inequalities: $O_k < O_{k'}$ and $T_k \geq T_{k'}$. The unloading order as well as the departure times of the containers that correspond to the vertices of a stable set are compatible, thereby they can be stored in a same stack if it has enough empty slots. In addition, there are at most H vertices in this stable set. So the number of containers assigned to the corresponding stack is inferior or equal to H. Therefore, a H-colouring corresponds to a valid assignment for the CSP. Similarly, it is easy to see that a solution of the CSP is a solution of the H-bounded colouring problem in the graph $G(N, O, T)$. We have the following lemma.

Lemma 2: Let $ICSP = (N, O, T, N_p, H, r, R, d)$ an instance of the container storage problem. The CSP has a solution for this instance if and only if the bounded H-colouring problem, considering the graph $G(N, O, T)$, has a solution.

Now, we give the main result of this section.

Theorem 3: The container storage problem is equivalent to the bounded colouring problem with minimum weight.

Proof: To establish this result, we prove that an instance of the CSP is equivalent to an instance of the BCP and vice versa. Let $ICSP = (N, O, T, N_p, H, r, R, d)$ an instance of the storage container problem and $G(N, O, T)$ the permutation graph associated to it. Consider $IBCP = (G(N, O, T), H, N_p, d)$ an instance of the BCP, which concerns the graph $G(N, O, T)$. N_p is the number of colours, H is the bound, and d is a matrix containing the weights. According to Lemma 2, a solution of the CSP is a solution of the BCP, and similarly a stack of the CSP corresponds to a colour of the BCP and vice versa. It follows then that the cost d_p^k of assigning a container $k \in N$ to the stack $p \in N_p$, is the same as the assignment of the vertex k to the colour corresponding to the stack p. So, the cost of a H-colouring of the graph $G(N, O, T)$ is similar to the cost of the solution of the corresponding CSP and vice versa. Therefore, we can find the optimal solution of the CSP if and only if we find the optimal solution of BCP. ■

According to Theorem 2, the bounded colouring problem is NP-hard in the case of permutation graphs if $H \geq 6$. Therefore, it follows from Theorem 3 that the CSP is NP-hard if $H \geq 6$.

Corollary 1: In the case where the containers have similar dimensions, the container storage problem is NP-hard if the maximum capacity of each stack is superior or equal to six.

If the containers have different sizes, we suppose that they are divided into groups, each having similar containers. Similarly, the stack are also divided into groups, each containing stacks that have equal measures. So, the CSP can be solved by considering separately several sub-problems, each consisting of allocating storage spaces to a group of containers that have similar sizes. For example, if there are three sizes of containers (20-feet, 40-feet, and 45-feet), we have three groups of container (K_{20}, K_{40}, K_{45}) and three groups of stack (P_{20}, P_{40}, P_{45}). In this example, to solve

the CPS, we can solve three sub-problems (CSP_{20}, CSP_{40}, CSP_{45}) by considering respectively (K_{20}, P_{20}), (K_{40}, P_{40}), and (K_{45}, P_{45}). So, the solutions of these sub-problems constitute the solution of the initial problem, as shown in Figure 4.

Since there is one size of container in each sub-problem, the corresponding graphs are permutation graphs (see Lemma 1). So, the container storage problem stills NP-hard even if the sizes of the containers are not similar.

Corollary 2: In the case where there are several dimensions of container, the container storage problem is NP-hard if the maximum number of containers allowed in a stack is superior or equal to six.

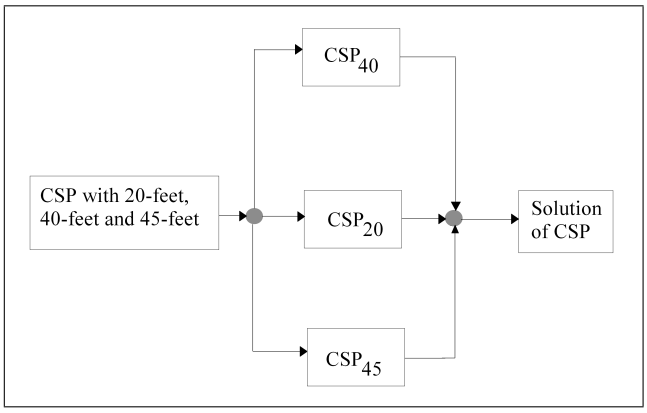


Figure 4. Case with three measures

C. Case of storage where some stacks are not empty at the beginning

In the case where some stacks are not empty at the beginning of the current storage period, the CSP can be formulated as a colouring problem with capacity (CPC), which is a generalization of the bounded colouring problem [21]. In the CPC, each colour p has a capacity c_p, which is the maximum number of nodes that can receive this colour. Different colours may have different capacities.

If we have an instance of the CSP, in which each stack has a capacity c_p, we can construct the corresponding graph and solve the CPC by considering that each colour can be used at most c_p times.

VI. Branch-and-cut algorithm

In the branch-and-cut algorithm, we consider the graph depicted in Section V-B, in which each node represents a container, and each edge connects two containers (nodes) that does not have similar dimensions or two containers that have conflicting arrival orders and departure times. So, as we know the fact that two adjacent containers cannot be stored into a same stack, we exploit this propriety in the branch-and-cut algorithm and we use the simplified mathematical model that follows.

The former decision variable $x^k_{p,i}$ is simplified and becomes x^k_p, which is defined as follows.

$$x^k_p = \begin{cases} 1 & \text{If the container k is assigned to the stack } p \\ 0 & \text{Otherwise} \end{cases}$$

As can be seen, this decision variable specifies the stack that is allocated to each container, but it does not point out the exact storage location of a container in a stack. This does not cause any problems, because the method that is used to construct the graph ensures that the unloading order and the departure times of the containers that are assigned to a same stack are compatible.

The new mathematical model is:

$$Minimize \sum_{k=1}^{N} \sum_{p=1}^{N_p} d^k_p x^k_p \tag{8}$$

The objective function (8) minimizes the total distance travelled by the straddle carriers between the ships and the container yard.

$$\sum_{p=1}^{N_p} x^k_p = 1, \quad \forall\, k = 1, ..., N \tag{9}$$

Constraint (9) requires that each container is assigned to a single stack.

$$x^k_p + x^{k'}_p \leq 1, \quad \forall\, (k, k') \in E,\ p = 1, ..., N_p \tag{10}$$

Constraint (10) ensures that the containers of each stack can be arranged following the same order that they were unloaded from ships, and the decreasing order of their departure times.

$$\sum_{k=1}^{N} x^k_p \leq c_p, \quad \forall\, p = 1, ..., N_p \tag{11}$$

Constraint (11) enforces that the capacity of each stack is not exceeded.

$$\sum_{k=1}^{N} x^k_p = 0, \quad \forall\, p = 1, ..., N_p : T_k > t_p \text{ or } R_k \neq r_p \tag{12}$$

Constraint (12) secures that each container can be assigned only to a compatible stack.

$$x^k_p \in \{0, 1\}, \quad \forall\, k = 1, ..., N,\ p = 1, ..., N_p \tag{13}$$

Constraint (13) states that the decision variables are boolean.

A. Description of the algorithm

The branch-and-cut algorithm is an exact resolution method, which combines the branch-and-bound method and the cutting plane method. Each of them proceeds by solving a sequence of relaxations of the mixed integer linear problem. The cutting plane methods improve the relaxation of a problem in order to ameliorate the approximation, while the branch-and-bound methods use a well known approach named "*divide and conquer*".

The branch-and-cut algorithm uses a search tree whose root node is the integer problem that needs to be solved. The

other nodes of the search tree are created sequentially by partitioning the search space, in other words, by creating new branches. The major difference between the branch-and-cut and the branch-and-bound is the fact that the former uses valid inequalities (cutting planes) to improve the solution found at each node of the search tree before performing branching. Notice that, a *valid inequality* is an inequality that must be satisfied by each solution of the mixed integer problem, but that is not necessarily satisfied by the solutions of the relaxations.

To perform a branch-and-cut, it is necessary to have these elements: one or several valid inequalities, a relaxation method, a technique to find an upper bound, a branching rule, and a method to look for violated valid inequalities (this method is called separation method).

In the following, we will explain them, before describing the algorithm.

1) Creation of a valid inequality: Let k a node (container) of the graph, such that $1 \leq k \leq N$. $N(k)$ the set of the neighbours of k, and p a colour (stack) such that $1 \leq p \leq N_p$.

After calculating the sum of constraints (10) in the neighbourhood $N(k)$, we get the following valid inequality named *neighbourhood inequality*.

$$\sum_{k' \in N(k)} x_p^{k'} + |N(k)| x_p^k \leq |N(k)| \qquad (14)$$

Proposition 1: For a solution of the integer program, the inequalities (10) and (14) are equivalent.

Proof: It suffices to prove that (14) implies (10), because the reverse is highlighted by the definition of (14).
As x_p^k is a binary variable, it can be equal to either 0 or 1.
• If $x_p^k = 1$, then $\sum_{k' \in N(k)} x_p^{k'} = 0$. Therefore, for all k' neighbour of k, we have $x_p^{k'} = 0$. Thereby, $x_p^k + x_p^{k'} = 1, \ \forall \ k' \in N(k)$.
• If $x_p^k = 0$, then $\sum_{k' \in N(k)} x_p^{k'} \leq |N(k)|$, which means that for all k' belonging to $N(k)$, $x_p^{k'}$ can be equal to either 0 or 1. Thus, $x_p^{k'} + x_p^k \leq 1, \ \forall \ k' \in N(k)$. ■

2) Relaxation of the problem: Generally, in a branch-and-cut algorithm, the integrity constraints (6) are relaxed by replacing them with ($x_p^k \geq 0, \ \forall \ k = 1, ..., N; \ p = 1, ..., N_p$). However, in our case, we go further by removing to the relaxed problem the constraint (10). This does not affect the optimality of the final solution, because at each node of the search tree, valid inequalities (14) are added to the program, in order to ensure the correctness of the solutions.

3) Preprocessing: The number of variables increases depending on the number of stacks (N_p) and the number of containers (N). In the case where all stacks were empty at the beginning of the storage period, if all containers are discharged from a same vessel, we can reduce the number of variables, because the containers are equidistant to the stacks. In such case, we only use the N stacks that are closer to the quay. This allows to significantly reduce the number of variables and to speed up the computation.

4) Upper bound: To find an upper bound, we solve the bounded vertex colouring problem applied on the graph defined in Section V-B. Each colour corresponds to a stack. We use a heuristic algorithm, which colours vertices one by one following the descending order of their number of uncoloured neighbours. For each vertex, it chooses among the admissible colours the one that fits to the nearest stack. The eligible colours are those that are not assigned to a vertex that is a neighbour of the considering vertex, and correspond to the stacks that are not full and satisfy the compatibility constraints (12). Whenever a vertex is coloured, the number of empty slot of the stack corresponding to the used colour is reduced.

5) Branching rule: We use the classical branching rule. At each node of the search tree, we create two branches by rounding the most fractional variable. Let x_p^k this variable. We put $x_p^k = 0$ in a branch, it means that the container k will not be assigned to the stack p in this branch. Then, in the other branch, we put $x_p^k = 1$, which means that the container k will be inevitably assigned to the stack p in this branch.

The most fractional variable is the one that is mostly half-way to the largest integer that is not greater than it and the smallest integer that is not lesser than it. For example, if we have $\{0.25, 0.45, 0.75\}$, the most fractional variable is 0.45.

To search the most fractional variable, we use this following algorithm.

```
1. select = 1;
2. For (k = 1, ..., N) do
    2.a. For (p = 1, ..., N_p) do
        2.a.1. If (|1/2 + ⌊x_p^k⌋ − x_p^k| < select) then
            select = x_p^k
        2.a.2 End If
    2.b. End For
3. End For
```

6) Separation method: At each node of the search tree, before creating branches, we use a simple algorithm to look for neighbourhood inequalities that are violated. To do this, we treat one by one each variable that is superior to 0.5 in the optimal solution of the current node. Let x_p^k one of these variables and S an integer that is initialized to zero. We calculate the number $|N(k)|$ of neighbours of the vertex k. Then, we add to S the value of x_p^k multiplied by $|N(k)|$. After that, we seek every variable $x_{p'}^{k'}$, which is such that k and k' are neighbours and $p = p'$, and we add the sum of their values to S. If $S > |N(k)|$, there is a violated inequality; therefore, we add to the sub-problem a constraint to avoid this.

7) Algorithm: The branch-and-cut algorithm that we propose to solve the container storage problem (BC-CSP) is as follows.

a. *Initialization:* We note P^0 the initial integer program. And then, we initialize the search tree $T = \{P^0\}$. After that, we use the algorithm depicted in Section VI-A3 to find an upper bound **UB**. If no solution is found, we set **UB** $= +\infty$.

b. *Stop criterion:* If $T = \emptyset$, the optimal solution is the one whose value equals **UB**. If **UB** $= +\infty$, there is no realizable

solution.

c. *Selection:* Choose a node P^l of T. This node will be removed from T after being explored.

d. *Relaxation:* Relax P^l, and then, solve it using the CPLEX solver. If there is no solution, set $\mathbf{LB}_l = +\infty$, after that, go to Step f. Otherwise, denominate $\mathbf{S}^{Rl}$ the optimal solution of the relaxation of P^l, and then, use its value to update $\mathbf{LB}_l$.

e. *Separation:* Seek all the valid inequalities that are violated by $\mathbf{S}^{Rl}$, add them to the relaxation of P^l, and then, go back to Step d.

f. *Fathoming and Pruning:* If $\mathbf{LB}_l \geq \mathbf{UB}$, then go back to Step b. If $\mathbf{LB}_l < \mathbf{UB}$ and $\mathbf{S}^{Rl}$ is a realizable integer solution, then update $\mathbf{UB} = \mathbf{LB}_l$ and remove from T every node j that satisfies $\mathbf{LB}_j \geq \mathbf{UB}$, after that, go back to Step b.

g. *Branching:* If $\mathbf{S}^{Rl}$ is fractional, then use the most fractional variable to perform a branching, in order to get two new sub-problems $P^{l,1}$ and $P^{l,2}$, after that, add them to T.

B. Example

Suppose that we need to store five containers in three empty stacks. The maximum number of containers allowed in a stack is equal to 3. We want to find the optimal storage plan, which does not lead to reshuffles and, which minimizes the total distance travelled by the straddle carriers between the quays and the storage yard. Table I and Table II show the distances and the characteristics of the containers, respectively. We consider that each container and each stack measures 20-feet.

TABLE I. Distances.

p	d_p^1	d_p^2	d_p^3	d_p^4	d_p^5
1	105	378	205	378	400
2	118	391	118	291	413
3	106	378	165	332	314

d_p^k is the distance between the stack p and the quay where is the container k.

TABLE II. Characteristics of the containers.

Container	*Departure time*	*Unloading order*
1	10	1
2	13	2
3	8	3
4	16	4
5	10	5

From these data we have the graph represented in Figure 5.

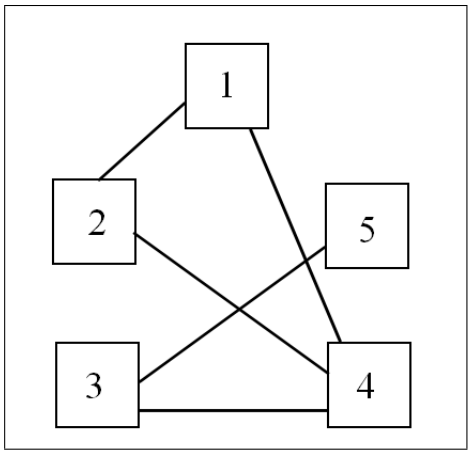


Figure 5. Graph constructed using the instance depicted in the Table II.

Firstly, we use the algorithm described in Section VI-A4 to look for an upper bound (in other words an integer solution that is not necessarily optimal). For this, we begin by the container 4, because it has most neighbours, we assign it to the stack 2, which is the nearest. After that, we remark that each of the remaining containers has one unassigned neighbour. Thereby, we realise these assignments, following the same order: container 1 to the stack 1, container 3 to stack 3, container 2 to stack 3, and container 5 to stack 1. The obtained upper bound is **UB**=1339, which equals $d_2^4 + d_1^1 + d_3^3 + d_3^2 + d_1^5$.

From the graph represented in Figure 5 and the data of Table I and Table II, we have the following linear integer program.

$$
P^0 = \begin{cases}
\text{Minimize} \\
d : 105x_1^1 + 118x_2^1 + 106x_3^1 + 378x_1^2 + 391x_2^2 + \\
\quad 378x_3^2 + 205x_1^3 + 118x_2^3 + 165x_3^3 + 378x_1^4 + \\
\quad 291x_2^4 + 332x_3^4 + 400x_1^5 + 413x_2^5 + 314x_3^5 \\
\text{subject to} \\
(1) : x_1^1 + x_2^1 + x_3^1 = 1 \\
(2) : x_1^2 + x_2^2 + x_3^2 = 1 \\
(3) : x_1^3 + x_2^3 + x_3^3 = 1 \\
(4) : x_1^4 + x_2^4 + x_3^4 = 1 \\
(5) : x_1^5 + x_2^5 + x_3^5 = 1 \\
(6) : x_1^1 + x_1^4 \leq 1 \\
(7) : x_2^1 + x_2^4 \leq 1 \\
(8) : x_3^1 + x_3^4 \leq 1 \\
(9) : x_1^1 + x_1^2 \leq 1 \\
(10) : x_2^1 + x_2^2 \leq 1 \\
(11) : x_3^1 + x_3^2 \leq 1 \\
(12) : x_1^2 + x_1^4 \leq 1 \\
(13) : x_2^2 + x_2^4 \leq 1 \\
(14) : x_3^2 + x_3^4 \leq 1 \\
(15) : x_1^3 + x_1^4 \leq 1 \\
(16) : x_2^3 + x_2^4 \leq 1 \\
(17) : x_3^3 + x_3^4 \leq 1 \\
(18) : x_1^3 + x_1^5 \leq 1 \\
(19) : x_2^3 + x_2^5 \leq 1 \\
(20) : x_3^3 + x_3^5 \leq 1 \\
(21) : x_1^1 + x_1^2 + x_1^3 + x_1^4 + x_1^5 \leq 3 \\
(22) : x_2^1 + x_2^2 + x_2^3 + x_2^4 + x_2^5 \leq 3
\end{cases}
$$

$$\begin{cases} (23): x_3^1 + x_3^2 + x_3^3 + x_3^4 + x_3^5 \leq 3 \\ (24): x_p^k \in \{0,1\},\ \forall\ k = 1,...,5;\ p = 1,2,3 \end{cases}$$

We initialize the search tree $T = \{P^0\}$. After that, we relax P^0, so we get R^0 that is as follows.

$$R^0 = \begin{cases} \text{Minimize} \\ d: 105x_1^1 + 118x_2^1 + 106x_3^1 + 378x_1^2 + 391x_2^2 + \\ \quad 378x_3^2 + 205x_1^3 + 118x_2^3 + 165x_3^3 + 378x_1^4 + \\ \quad 291x_2^4 + 332x_3^4 + 400x_1^5 + 413x_2^5 + 314x_3^5 \\ \text{subject to} \\ (1): x_1^1 + x_2^1 + x_3^1 = 1 \\ (2): x_1^2 + x_2^2 + x_3^2 = 1 \\ (3): x_1^3 + x_2^3 + x_3^3 = 1 \\ (4): x_1^4 + x_2^4 + x_3^4 = 1 \\ (5): x_1^5 + x_2^5 + x_3^5 = 1 \\ (21): x_1^1 + x_1^2 + x_1^3 + x_1^4 + x_1^5 \leq 3 \\ (22): x_2^1 + x_2^2 + x_2^3 + x_2^4 + x_2^5 \leq 3 \\ (23): x_3^1 + x_3^2 + x_3^3 + x_3^4 + x_3^5 \leq 3 \\ (24): x_p^k \geq 0,\ \forall\ k = 1,...,5;\ p = 1,2,3 \end{cases}$$

After that, we solve R^0, using the CPLEX solver (version 12.5). And then, we get ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 1, x_2^2 = 0, x_1^3 = 0, x_2^3 = 1, x_1^4 = 0, x_2^4 = 1, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0, x_3^2 = 0, x_3^3 = 0, x_3^4 = 0, x_3^5 = 1$), which equals 1206. This is not a realizable solution, because it does not satisfy the following valid inequalities.

$(25): x_1^2 + x_1^4 + 2x_1^1 <= 2$
$(26): x_1^1 + x_1^4 + 2x_1^2 <= 2$
$(27): x_2^4 + x_2^5 + 2x_2^3 <= 2$
$(28): x_2^1 + x_2^2 + x_2^3 + 3x_2^4 <= 3$

Then, we add the inequalities (25), (26), (27), and (28) to R^0. After that, we solve it again, and then, we obtain ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 0, x_2^2 = 0, x_1^3 = 0, x_2^3 = 0.6, x_1^4 = 0, x_2^4 = 0.8, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0, x_3^2 = 1, x_3^3 = 0.4, x_3^4 = 0.2, x_3^5 = 0$), which equals 1233. This solution is fractional. In addition, it does not satisfy the following valid inequalities.

$(29): x_3^1 + x_3^4 + 2x_3^2 <= 2$
$(30): x_3^3 + x_3^5 <= 1$

We add the inequalities (29) and (30) to R^0. And then, we solve it again. So, we get ($x_1^1 = 0.666666666666667, x_2^1 = 0, x_1^2 = 0.666666666666667, x_2^2 = 0, x_1^3 = 0, x_2^3 = 1, x_1^4 = 0, x_2^4 = 0, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0.333333333333333, x_3^2 = 0.333333333333333, x_3^3 = 0, x_3^4 = 1, x_3^5 = 1$), which equals 1247.33333333333. This solution violates the following valid inequality.

$(31): x_3^1 + x_3^2 + x_3^3 + 3x_3^4 <= 3$

After adding the inequality (31) to R^0, we solve it again. And then, we get ($x_1^1 = 0.75, x_2^1 = 0, x_1^2 = 0.5, x_2^2 = 0, x_1^3 = 0.125, x_2^3 = 0.875, x_1^4 = 0, x_2^4 = 0.25, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0.25, x_3^2 = 0.5, x_3^3 = 0, x_3^4 = 0.75, x_3^5 = 1$), which equals 1247.875. This solution does not violate any valid inequalities, but it is not integer. So, we use the variable x_1^2 to do a branching. And then, we get two sub-problems $P^{00} = P^0 \cup \{x_1^2 = 0\}$ and $P^{01} = P^0 \cup \{x_1^2 = 1\}$. After that, we add the new sub-problems to T, and we remove P^0. So, we have $T = \{P^{00}, P^{01}\}$.

Since T is not yet empty, we continuous the algorithm. So, we select P^{00} from T. After that, we relax it, and we obtain R^{00}. Then, we solve R^{00} using CPLEX, and we obtain ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 0, x_2^2 = 0.125, x_1^3 = 0.375, x_2^3 = 0.625, x_1^4 = 0, x_2^4 = 0.75, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0, x_3^2 = 0.875, x_3^3 = 0, x_3^4 = 0.25, x_3^5 = 1$), which equals 1250.5. This solution does not violate any valid inequalities. So, we use x_1^3 to do a branching. Then, we get two new sub-problems $P^{000} = P^{00} \cup \{x_1^3 = 0\}$ and $P^{001} = P^{00} \cup \{x_1^3 = 1\}$. We add these sub-problems to T, and we remove P^{00}. So, we have $T = \{P^{01}, P^{000}, P^{001}\}$.

We select P^{01}. After that, we relax and solve it. And then, we obtain ($x_1^1 = 0, x_2^1 = 0, x_1^2 = 1, x_2^2 = 0, x_1^3 = 0.166666666666667, x_2^3 = 0.833333333333333, x_1^4 = 0, x_2^4 = 0.333333333333333, x_1^5 = 0, x_2^5 = 0, x_3^1 = 1, x_3^2 = 0, x_3^3 = 0, x_3^4 = 0.666666666666667, x_3^5 = 1$), which equals 1248.83333333333. This solution violates the following valid inequality.

$(32): x_3^2 + x_3^4 + 2x_3^1 <= 1$

We add the inequality (32) to R^{01}. And then, we solve it again. After that, we get ($x_1^1 = 0, x_2^1 = 0.125, x_1^2 = 1, x_2^2 = 0, x_1^3 = 0.375, x_2^3 = 0.625, x_1^4 = 0, x_2^4 = 0.75, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0.875, x_3^2 = 0, x_3^3 = 0, x_3^4 = 0.25, x_3^5 = 1$), which equals 1251.375. This solution does not violate any valid inequalities. So, we use x_1^3 to do a branching. Two sub-problems are then created and added to T, $P^{010} = P^{01} \cup \{x_1^3 = 0\}$ and $P^{011} = P^{01} \cup \{x_1^3 = 1\}$. We remove P^{01} from T, which becomes $T = \{P^{000}, P^{001}, P^{010}, P^{011}\}$.

We select P^{000}. And then, we relax and solve it. After that, we get ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 0, x_2^2 = 0.375, x_1^3 = 0, x_2^3 = 0.875, x_1^4 = 0, x_2^4 = 0.25, x_1^5 = 0.125, x_2^5 = 0, x_3^1 = 0, x_3^2 = 0.625, x_3^3 = 0.125, x_3^4 = 0.75, x_3^5 = 0.875$), which equals 1258.25. This solution does not violate any valid inequalities. So, we use x_2^2 to perform a branching. Then, we get two new sub-problems $P^{0000} = P^{000} \cup \{x_2^2 = 0\}$ and $P^{0001} = P^{000} \cup \{x_2^2 = 1\}$. We remove P^{000} from T. After that, we add to T the new sub-problems. So, we have $T = \{P^{001}, P^{010}, P^{011}, P^{0000}, P^{0001}\}$.

We select P^{001}. After that, we relax and solve it. Then, we get ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 0, x_2^2 = 0, x_1^3 = 1, x_2^3 = 0, x_1^4 = 0, x_2^4 = 1, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0, x_3^2 = 1, x_3^3 = 0, x_3^4 = 0, x_3^5 = 1$), which equals 1293. This solution is integer and does not violate any valid inequalities. So, we update the upper bound **UB**=1293. And then, we remove P^{001} from T, which becomes $T = \{P^{010}, P^{011}, P^{0000}, P^{0001}\}$.

We select P^{010}. Then, we relax and solve it. After that, we get ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 0, x_2^2 = 0, x_1^3 = 1, x_2^3 = 0, x_1^4 = 0, x_2^4 = 1, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0, x_3^2 = 1, x_3^3 = 0, x_3^4 = 0, x_3^5 = 1$), which equals 1293. This is an integer solution, which does not violate any valid inequalities. We remove

P^{010} from T, which becomes $T = \{P^{011}, P^{0000}, P^{0001}\}$.

We select P^{011}. After that, we relax and solve it. Then, we get ($x_1^1 = 0, x_2^1 = 0, x_1^2 = 1, x_2^2 = 0, x_1^3 = 1, x_2^3 = 0, x_1^4 = 0, x_2^4 = 1, x_1^5 = 0, x_2^5 = 0, x_3^1 = 1, x_3^2 = 0, x_3^3 = 0, x_3^4 = 0, x_3^5 = 1$), which equals 1294. We remove P^{011} from T, because it gives a solution that has a value superior to the upper bound. So, we have $T = \{P^{0000}, P^{0001}\}$.

We select P^{0000}. After that, we relax and solve it. Then, we get ($x_1^1 = 0.875, x_2^1 = 0.125, x_1^2 = 0, x_2^2 = 0, x_1^3 = 0, x_2^3 = 0.625, x_1^4 = 0.25, x_2^4 = 0.75, x_1^5 = 0.375, x_2^5 = 0, x_3^1 = 0, x_3^2 = 1, x_3^3 = 0.375, x_3^4 = 0, x_3^5 = 0.625$), which equals 1279.25. This solution does not violate any valid inequalities. So, we use x_2^3 to do a branching. Then, we get two new sub-problems: $P^{00000} = P^{0000} \cup \{x_2^3 = 0\}$ and $P^{00001} = P^{0000} \cup \{x_2^3 = 1\}$. After that, we update the list of active nodes, $T = \{P^{0001}, P^{00000}, P^{00001}\}$.

We select P^{0001}. Then, we relax and solve it. After that, we get ($x_1^1 = 1, x_2^1 = 0, x_1^2 = 0, x_2^2 = 1, x_1^3 = 0, x_2^3 = 1, x_1^4 = 0, x_2^4 = 0, x_1^5 = 0, x_2^5 = 0, x_3^1 = 0, x_3^2 = 0, x_3^3 = 0, x_3^4 = 1, x_3^5 = 1$), which equals 1260. This solution is integer and does not violate any valid inequalities. So, we update the upper bound, **UB**=1260. And then, we remove P^{0001}, P^{00000}, and P^{00001} from T. Notice that P^{00000} and P^{00001} are not explored because their lower bound (1279.25) is superior to the general upper bound (1260). This is the end of the algorithm, because $T = \emptyset$.

The search tree corresponding to that example is represented in Figure 6. The red nodes are those that are pruned, while the green nodes are the ones that allow to update the upper bound (in other words those that give integer solution, which is better than the current one). The order, in which the nodes are explored is specified by the blue writings.

VII. Ant colony algorithm

The first ant colony algorithm is invented by Dorigo et al. [22]. They were inspired by the behaviour of the natural ants when they are looking for food. These animals communicate indirectly via a natural substance named pheromone, in order to discover the shortest path between their anthill and a location where there is food. This substance is continuously deposited on the travelled ways. Therefore, since the short paths lead more quickly to the food, the pheromone will be accumulated there more quickly. So, they will be more preferable. In addition to this, the pheromone tends to disappear on the longer paths due to the evaporation.

To apply an ant colony algorithm to a problem, it is necessary to define how to represent a solution. So, in the following, we firstly specify how we encode our solutions, before detailing the ant colony algorithm that we propose to solve the container storage problem.

A. Method to represent a solution

In the ant colony algorithm, we represent a solution as an array that has two rows. The containers are written in the first row, while the stacks are noted in the second. The number of columns in the solution is equal to the number of containers that need to be stored. The following example represents a solution, in which six containers are assigned to three stacks.

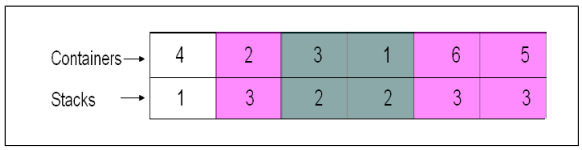


Figure 7. Example of solution

This solution corresponds to the following assignment

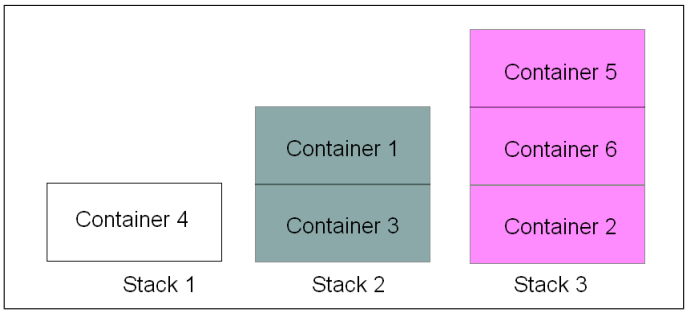


Figure 8. Stacking manner

If several containers are assigned to a same stack, they will be stored following the increasing order of their column numbers. This enables to take into account the arrival order constraint (5) and the departure times constraints (6) of the first mathematical model. For example, the containers 3 and 1 are both assigned to the stack 2, but the container 3 has the lowest storage location.

B. Algorithm

In the ant colony algorithm, which we propose to solve the container storage problem (ANTCSP), we use four parameters, which are: the number of ants (NA), the number of iterations (NT_{Max}), the minimum threshold of pheromone (τ_{Min}), and the maximum threshold of pheromone (τ_{Max}). This is based on the **Min-Max** version of the ant colony algorithm, more informations about the different versions of ant colony algorithms are available in [23]. The ant colony algorithm progresses as follows:

1: Initialization of pheromone.
2: Construction of a solution by each ant.
3: Evaluation of the solutions.
4: Initialization of the number of iterations, $NT = 1$.
5: While ($NT < NT_{Max}$) do:
    5.a: Update pheromone.
    5.b: Construction of new solutions by the ants.
    5.c: Evaluation of the solutions.
    5.d: $NT = NT + 1$.
End while.

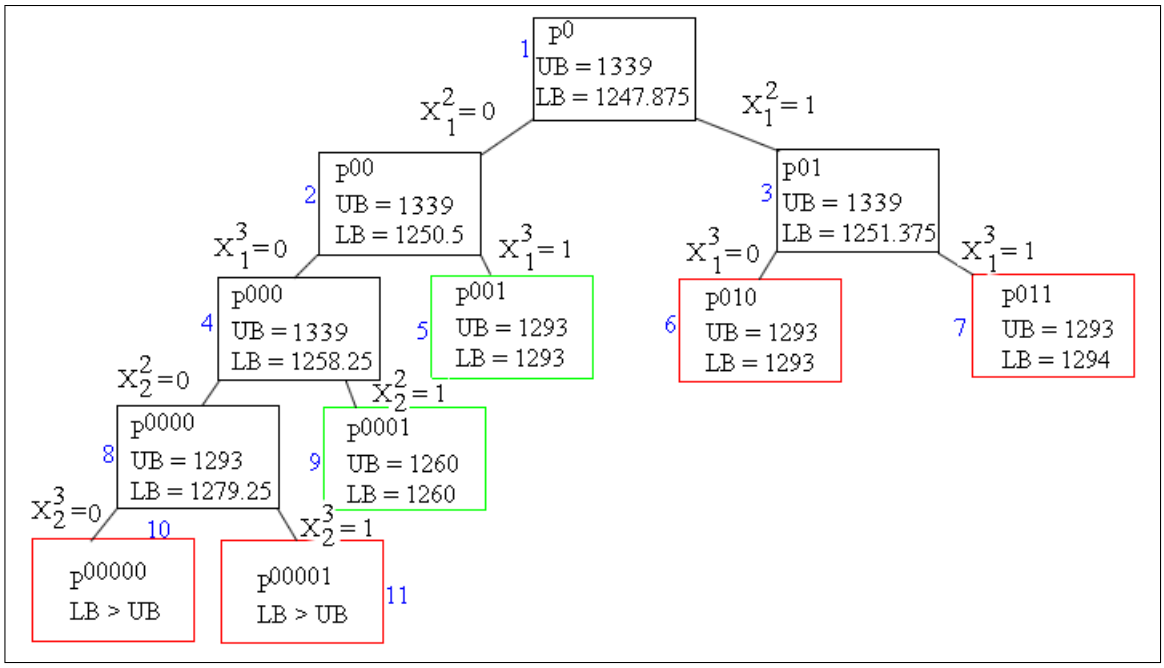


Figure 6. Search tree.

1) Construction method of a solution by an ant: The construction of a solution is done sequentially by adding successively elements. Therefore, before beginning the search of solutions, we firstly construct the set of options. This means, the set of couples (container: k, stack: p) that are compatible. In other words, every pair (k, p) that satisfies these three conditions:

- $c_p > 0$, (the stack p is not full)
- $r_p = R_k$, (the container k and the stack p have similar dimensions)
- $t_p \geq T_k$, (the departure time of the container k is inferior to that of the container that is already at the top of the stack p at the beginning of the new storage period).

All these couples form the set of options (for commodity, we name it E), and each option (k, p) has a pheromone trail, which is initialized ($\tau(k, p) = \tau_{Max}$).

In the step 2 of the ant colony algorithm, we use this following pseudo-code to construct a solution.

1: Let $S = \emptyset$, the solution that is being built.
2: $E_1 = E$.
3: Choose arbitrarily an element of E_1 and add it to S.
4: While (E_1 is not empty) do:
  4.a: Update E_1.
  4.b: Calculate the probability $P_{(k,p)}$ of every element (k, p) remaining in E_1.

$$P_{(k,p)} = \frac{(\tau_{(k,p)})^{\alpha} \times (\frac{1}{d_p^k})^{\beta}}{\sum_{(k,p)\in E_1} (\tau_{(k,p)})^{\alpha} \times (\frac{1}{d_p^k})^{\beta}}$$

  Where α and β are positive real numbers inferior to 1, and $\frac{1}{d_p^k}$ is the visibility.
  4.c: Choose the element of E_1 that has the largest probability and add it to S.
  End while.
5: If (the number of couples belonging to S is inferior to N) then
  5.a: Go back to Step 1.

Whenever a couple is added to the solution S, we remove it from the set of options E_1. After that, we decrease the capacity of the corresponding stack. And then, we delete from the set of options every couple that may compromise the validity of the solution. For example, suppose that we add to S the option (k, p). So, we update the capacity of the stack p (this means $c_p = c_p - 1$), and we remove from E_1 all couple (k', p') that satisfies at least one of these four conditions:

- $c_{p'} = 0$, (the stack is full)
- $k' = k$, (the container is already assigned)
- $O_k > O_{k'}$, (incompatible unloading numbers)
- $T_k < T_{k'}$, (incompatible departure times).

2) Method to update the pheromone trails: At the end of each iteration, the pheromone trails are updated in two steps. Firstly, an evaporation decreases the pheromone of each option, like follows:

$$\forall\ (k, p) \in E, \quad \tau_{(k,p)} = (1 - \rho)\tau_{(k,p)}$$

Where ρ is the evaporation rate, and $0 < \rho < 1$.
If $\tau_{(k,p)} < \tau_{Min}$, we adjust it ($\tau_{(k,p)} = \tau_{Min}$).

Unlike the evaporation, the augmentation of the pheromone trails is done only on the couples that belong to the best solution found during the current iteration. Let S_{bc} that solution, the pheromone trails of its couples are increased as follows:

$$\forall\ (k, p) \in S_{bc}, \quad \tau_{(k,p)} = \tau_{(k,p)} + \frac{1}{|O_{bc} - O_b + 1|}$$

Where O_b is the value of the best solution found since the beginning of the algorithm until the current iteration, and O_{bc} is the value of S_{bc}. If $\tau_{(k,p)} > \tau_{Max}$, we adjust it ($\tau_{(k,p)} = \tau_{Max}$).

VIII. Hybrid ant colony and branch-and-cut algorithm

In the hybrid ant colony and branch-and-cut algorithm (HACBC), we use the ant colony algorithm to find an upper bound (**UB**) of the container storage problem. This upper bound is then used in the branch-and-cut algorithm, in order to accelerate it by only exploring the nodes that have lower bounds inferior to **UB**. The Hybrid algorithm is represented in Figure 9, where P^0 represents the initial integer problem and S designates the current best integer solution. The starter step and the final step are coloured in pink.

As can be seen in Figure 9, the HACBC algorithm is stopped only if the list of active nodes becomes empty. After the initialization of **UB** and S, the search tree is initialized with P^0, and then, these following actions are repeated in the same order until the stop condition:

- Select an element P^j in T.
- Relax and solve P^j with CPLEX.
- While there are violated valid inequalities, add them to the relaxation of P^j, and then, solve it again.
- If the solution of P^j is integer and better than S, update **UB** and S, remove from T every node that has a lower bound superior or equal to **UB**.
- If the solution of P^j is fractional and has a value inferior to **UB**, use the most fractional variable to perform a branching.
- Remove P^j from T.

IX. Numerical results

In this section, we present the numerical results of the different algorithms that are proposed in this paper. The experiments were performed using a computer DELL PRECISION T3500 Intel Xeon 5 GHz processor. Each algorithm is implemented in C++ language. In addition, we use the CPLEX solver version 12.5, and the framework SCIP that is very useful because it allows a total control over the different components of the branch-and-cut algorithm.

The details concerning the data used in the numerical simulations are noted in Table III.

TABLE III. Benchmark set-up

Number of containers	$50 \leq N \leq 1400$
Number of stacks	$100 \leq N_p \leq 3500$
Maximum height of a stack	3
Percentage of vacant storage locations	$50\% \leq \frac{100 \times \sum_{p=1}^{N_p} c_p}{3 \times N_p}$
Number of sizes of containers	3 sizes: 20 feet, 40 feet, and 45 feet
Average dwell time of a container	4 days
Distance between the stack p and the quay where is the container k	$300\ m \leq d_p^k \leq 800\ m$

Before comparing the performances of the algorithms, we firstly researched the best values of the ant colony algorithm's parameters. To do this, we treat them individually. At each step, we vary the value of one parameter, the values of the other parameters do not change during the iterations. We applied this method on different instances, for each parameter, and then, we obtain the results described in Table IV.

To look for the suitable number of iterations (NT_{Max}), we considered the integers that are between 20 and 100, and we choose the number, from which the objective function does not decrease any more. Similarly, the number of ants (NA) are searched between 10 and 100. As for the exponent of the pheromone (α), the exponent of the visibility (β), and the rate of pheromone evaporation (ρ), they are dealt with by considering de real numbers that are between 0 and 1. And finally, the minimum threshold of pheromone (τ_{Min}) is sought by considering the integers that are between 1 and 5, while the maximum threshold of pheromone (τ_{Max}) is looked for by considering the integers that are between 5 and 10.

TABLE IV. Values of the ant colony algorithm's parameters.

Parameter	Value
Number of iterations	$NT_{Max} = 40$
Number of ants	$NA = 17$
Exponent of the pheromone	$\alpha = 0.3$
Exponent of the visibility	$\beta = 0.2$
Rate of pheromone evaporation	$\rho = 0.2$
Minimum threshold of pheromone	$\tau_{Min} = 1$
Maximum threshold of pheromone	$\tau_{Max} = 10$

Unlike the ant colony algorithm, the branch-and-cut algorithm and the hybrid algorithm give optimal results. However, the ant colony algorithm gives good upper bounds as can be seen in Table V, where the values of the objective function are mentioned for twenty four instances.

gap is the percentage of deviation, it is calculated using the following formula:

$$gap = \frac{Obj_{(ANTCSP)} - Obj_{(optimal)}}{Obj_{(optimal)}} \times 100$$

$Obj_{(ANTCSP)}$ is the value of the solution found by the ant colony algorithm, and $Obj_{(optimal)}$ is the value of the optimal solution found by the CPLEX solver. For the instances that could not be solved by CPLEX, $Obj_{(optimal)}$ represents the value of the optimal solution found by HACBC.

val is the value of the objective function.

The symbol $\cdots$ means that the execution is interrupted because it lasted more than 3 hours. Similarly, the symbol — means that the computer memory is insufficient to enable the resolution of the instance.

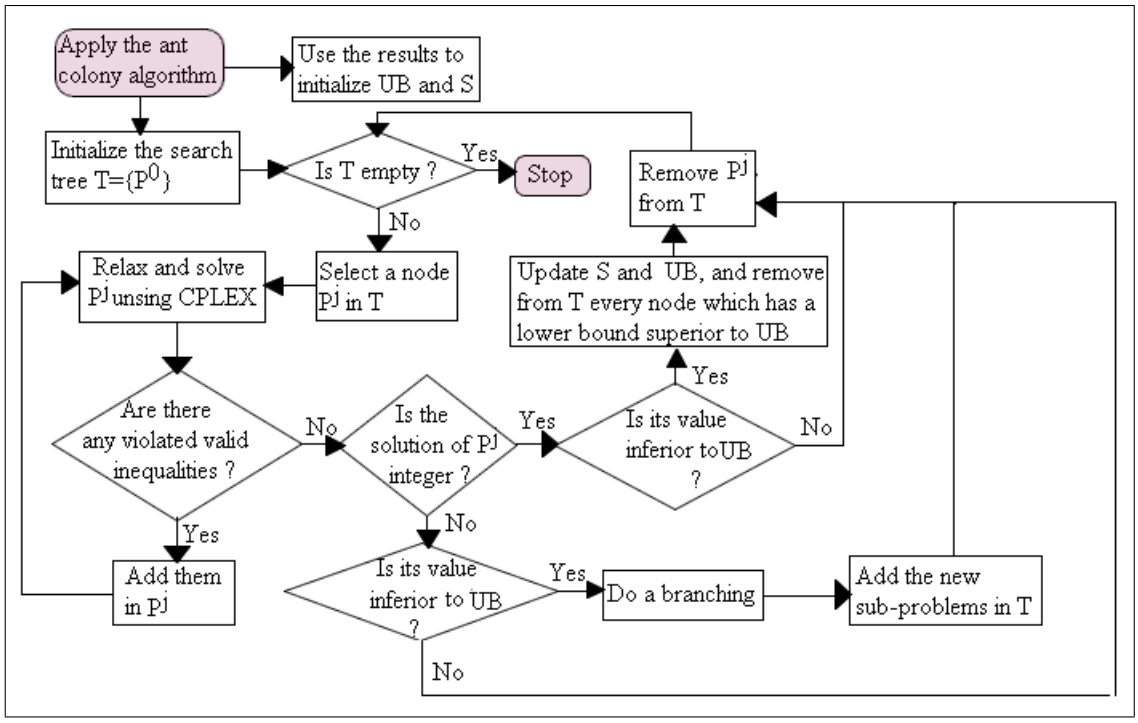


Figure 9. HACBC.

TABLE V. Comparison of the algorithms' solutions

N	N_p	ANTCSP		CPLEX	BC-CSP	HACBC
		val	gap	val	val	val
100	500	53682	0.85%	53230	53230	53230
150	500	62042	3.09%	60483	60483	60483
100	700	46672	4.02%	44867	44867	44867
100	1500	53945	1.81%	52988	52988	52988
50	200	15882	0%	15882	15882	15882
200	200	71682	1.43%	70671	70671	70671
80	100	25608	0%	25608	25608	25608
90	100	34636	0%	34636	34636	34636
100	100	39026	4.5%	37168	37168	37168
150	200	55388	0.69%	55011	55011	55011
100	3500	33035	1.5%	—	32547	32547
200	3500	67604	1.97%	—	66300	66300
300	3500	92406	1.57%	—	90979	90979
400	3500	125145	1.38%	—	123438	123438
500	3500	171137	1.33%	—	168895	168895
600	3500	184924	0.82%	—	183415	183415
700	3500	216424	0.6%	—	215138	215138
800	3500	245357	0.59%	—	243917	243917
900	3500	276831	0.95%	—	274238	274238
1000	3500	323187	0.87%	—	320415	320415
1100	3500	338955	0.48%	—	337347	337347
1200	3500	375269	0.47%	—	373498	373498
1300	3500	405622	0.64%	—	403054	403054
1400	3500	432816	0.39%	—	...	431116

The results depicted in Table V show that, in some cases, the ant colony algorithm gives optimal results. In addition, it gives generally good results that have percentages of deviation inferior to 5%.

In Table VI, we compare the execution times of CPLEX, BC-CSP, and HACBC. The ant colony algorithm is fast but as it does not give optimal results every time, it would not be relevant to compare its execution times to those of the other algorithms.

TABLE VI. Comparison of the execution times

N	N_p	HACBC	BC-CSP	CPLEX
100	500	0 sec	0 sec	2 min 58 sec
150	500	0 sec	1 sec	15 min 45 sec
100	700	0 sec	0 sec	4 min 11 sec
100	1500	0 sec	0 sec	11 min 16 sec
50	200	0 sec	0 sec	2 sec
200	200	1 sec	3 sec	14 min
80	100	0 sec	0 sec	1 min 5 sec
90	100	0 sec	0 sec	1 min 29 sec
100	100	0 sec	0 sec	2 min 11 sec
150	200	0 sec	1 sec	53 min 50 sec
100	3500	0 sec	0 sec	—
200	3500	0 sec	3 sec	—
300	3500	6 sec	14 sec	—
400	3500	11 sec	41 sec	—
500	3500	35 sec	1 min 36 sec	—
600	3500	2 min 35 sec	6 min 13 sec	—
700	3500	1 min 46 sec	5 min 57 sec	—
800	3500	3 min 20 sec	9 min 49 sec	—
900	3500	6 min 14 sec	15 min 35 sec	—
1000	3500	33 min 40 sec	1 h 41 min 26 sec	—
1100	3500	34 min 5 sec	1 h 43 min 47 sec	—
1200	3500	51 min 12 sec	2 h 5 min 4 sec	—
1300	3500	51 min 33 sec	2 h 21 min 20 sec	—
1400	3500	1 h 3 sec	...	—

As can be seen in Table VI, our branch-and-cut algorithm is quicker than the CPLEX solver version 12.5 for the resolution of the container storage problem. However, these two methods are outperformed by the hybrid ant colony and branch-and-cut algorithm.

The results that are contained in Table V and Table VI are obtained by doing a single execution for each instance.

X. Conclusion and Future Work

This paper deals with the inbound container storage problem at seaport terminal. A container terminal that uses straddle

carriers as handling and transfer equipments is considered. A mathematical model, which determines an accurate storage location for each container is proposed. This mathematical model takes into account physical and operational constraints, like the order, in which the containers are unloaded from vessels, and minimizes the total distance travelled by the straddle carriers between the quays and the container yard, in order to shorten the berthing times of the ships. A demonstration of the NP-hardness of the container storage problem is given. For the numerical resolution, we propose an efficient hybrid ant colony and branch-and-cut algorithm. This hybridization allows to improve the performances of the branch-and-cut algorithm that was proposed in [1]. In addition, it is an exact resolution method and is able to solve quickly large instances, which cannot be solved by the CPLEX solver.

In the future, we plan to test other branching rules and other valid inequalities. We also plan to study the case of container terminals that use automatic equipments like automatic guided vehicles and rail mounted cranes, and to propose other efficient resolution methods.

REFERENCES

[1] N. F. Ndiaye, A. Yassine, and I. Diarrassouba, "A Branch-and-Cut Algorithm to Solve the Container Storage Problem," in Proceedings of the ninth international conference on systems (ICONS), February 23–27, 2014, Nice, France, 2014, ISBN: 978-1-61208-319-3, ISSN: 2308-4243, URL: http://www.thinkmind.org/ [accessed: 2014-07-27].

[2] C. Zhang, J. Liu, Y. W. Wan, K. G. Murty, and R. J. Linn, "Storage space allocation in container terminals," Transportation Research Part B: Methodological, vol. 37, 2003, pp. 883–903.

[3] M. Bazzazi, N. Safaei, and N. Javadian, "A genetic algorithm to solve the storage space allocation problem in a container terminal," Computers & Industrial Engineering, vol. 56, 2009, pp. 44–52.

[4] C. Park and J. Seo, "Mathematical modeling and solving procedure of the planar storage location assignment problem," Computers & Industrial Engineering, vol. 57, 2009, pp. 1062–1071.

[5] D.-H. Lee, X. J. Cao, Q. Shi, and J. H. Chen, "A heuristic algorithm for yard truck scheduling and storage allocation problems," Transportation Research Part E: Logistics and Transportation Review, vol. 45, 2009, pp. 810–820.

[6] E. Kozan and P. Preston, "Mathematical modeling of container transfers and storage locations at seaport terminals," OR Spectrum, vol. 28, 2006, pp. 519–537.

[7] S. Sauri and E. Martin, "Space allocating strategies for improving import yard performance at marine terminals," Transportation Research Part E: Logistics and Transportation Review, vol. 47, 2011, pp. 1038–1057.

[8] K. H. Kim and H. B. Kim, "Segregating space allocation models for container inventories in port container terminals," International Journal of Production Economics, vol. 59, 1999, pp. 415–423.

[9] C. Jinxin, S. Qixin, and D.-H. Lee, "A Decision Support Method for Truck Scheduling and Storage Allocation Problem at Container," Tsinghua Science & Technology, vol. 13, 2008, pp. 211–216.

[10] M. Yu and X. Qi, "Storage space allocation models for inbound containers in an automatic container terminal," European Journal of Operational Research, vol. 226, 2013, pp. 32–45.

[11] R. Moussi, N. F. Ndiaye, and A. Yassine, "Hybrid Genetic Simulated Annealing Algorithm (HGSAA) to Solve Storage Container Problem in Port," Intelligent Information and Database Systems, Lecture Notes in Computer Science, vol. 7197, 2012, pp. 301–310.

[12] K. H. Kim and K. Y. Kim, "Optimal price schedules for storage of inbound containers," Transportation Research Part B: Methodological, vol. 41, 2007, pp. 892–905.

[13] P. Preston and E. Kozan, "An approach to determine storage locations of containers at seaport terminals," Computers & Operations Research, vol. 28, 2001, p. 983995.

[14] K. H. Kim, Y. M. Park, and K.-R. Ryu, "Deriving decision rules to locate export containers in container yards," European Journal of Operational Research, vol. 124, 2000, pp. 89–101.

[15] L. Chen and Z. Lu, "The storage location assignment problem for outbound containers in a maritime terminal," International Journal of Production Economics, vol. 135, 2012, pp. 73–80.

[16] Y. J. Woo and K. H. Kim, "Estimating the space requirement for outbound container inventories in port container terminals," International Journal of Production Economics, vol. 133, 2011, pp. 293–301.

[17] K. H. Kim and K. T. Park, "A note on a dynamic space-allocation method for outbound containers," European Journal of Operational Research, vol. 148, 2003, p. 92101.

[18] N. Nishimura, A. Imai, G. K. Janssens, and S. Papadimitriou, "Container storage and transshipment marine terminals," Transportation Research Part E: Logistics and Transportation Review, vol. 45, 2009, p. 771786.

[19] B. Dushnik and E. W. Miller, "Partially ordered sets," American Journal of Mathematics, vol. 63, 1941, pp. 600–610.

[20] K. Jansen, "The mutual exclusion scheduling problem for permutation and comparability graphs," STACS 98, Lecture Notes in Computer Science, vol. 1373, 1998, pp. 287–297.

[21] F. Bonomo, S. Mattia, and G. Oriolo, "Bounded coloring of co-comparability graphs and the pickup and delivery tour combination problem," Theoretical Computer Science, vol. 412, 2011, pp. 6261–6268.

[22] M. Dorigo, V. Maniezzo, and A. Colorni, "The Ant System: Optimization by a colony of cooperating agents," Transactions on Systems, Man, and Cybernetics-Part B, IEEE, vol. 26, 1996, pp. 1–13.

[23] M. Dorigo and T. Stützle, "The Ant Colony Optimization Metaheuristic: Algorithms, Applications, and Advances," Handbook of Metaheuristics, International Series in Operations Research & Management Science, vol. 57, 2003, pp. 250–285.

Team Assistance in a Software Engineering Team: A Field Study

Pierre N. Robillard, Sébastien Cherry,
Department of Computer and Software Engineering
Polytechnique Montréal
Montréal, Canada
pierre.robillard@polymtl.ca
sebastien.cherry@polymtl.ca

François Chiocchio, Carolyne Hass
Department of Psychology
Université de Montréal
Montréal, Canada
chiocchio@telfer.uottawa.ca
Carolyne.hass@umontreal.ca

***Abstract*— Physically collocated teammates often interact spontaneously while working solo on their assigned tasks. These ad hoc interactions could be perceived as counterproductive when they are seen as interruptions or they could be perceived as productive when they are seen as ad hoc team assistances, which contribute to the team awareness, trust amongst team members, and improved shared mental model. This paper reports on a field study performed in a professional environment. Team activities have been continuously video recorded over a period of two months. More than 400 ad hoc interactions have been analyzed. Ad hoc interactions required up to 30% of the team total time. These ad hoc interactions involve all the team members and as such may contribute to team awareness and improvement of shared mental model. Ad hoc team assistances can be categorized according to two purposes: the application domain or the development environment. This study shed light on the team dynamics of collocated teams and can provide insight into the challenges faced by the distributed software development teams. Suggestions are formulated for the management of team assistance activities.**

Keywords- Team process, team assistance, field study, ad hoc interactions, collocated team.

I. INTRODUCTION

Software development paradigms involved harnessing teammate interactions, which are the core of any team activities. Previous studies present overviews of the types of knowledge exchanges occurring during team interactions [1] and explore the roles of ad hoc interactions and the social side of software engineering [2]. Whatever the approach, members of ongoing collocated team engage in teamwork and taskwork. While teamwork refers to how team members work to combine their thoughts, actions, and feelings to coordinate and adapt, and to reach a common goal, taskwork refers to how team members interact individually with tasks, tools, machines, and systems [3][4]. Teamwork is often performed synchronously during scheduled or planned meetings, where team members interact in a shared activity. Typical examples are brain-storming sessions and design reviews [5]. Taskworks occur when teammates work solo on their assigned tasks. In a collocated software development environment, the task is often related to programming, debugging, or testing activities. During these solo activities teammates will nevertheless interact on an ad hoc basis.

Organizational psychologists have long been interested in the dynamics of team interactions and it may be wise for software engineers to capitalize on their expertise to better understand the dynamics of software development team. There is 50-year long tradition of studying helping behaviors in the industrial and organizational psychology literature [6][7]. These helping behaviors refer to a larger class of behaviors called Team Assistances, which could be defined as "individual behavior that is discretionary, not directly or explicitly recognized by the formal reward system, and that in the aggregate promotes the effective functioning of the organization" [8].

"By discretionary, we mean that the behavior is not an enforceable requirement of the role or the job description, that is, the clearly specifiable terms of the person's employment contract with the organization; the behavior is rather a matter of personal choice, such that its omission is not generally understood as punishable." [8]. According to Borman's model [9] of organizational citizenship performance these behaviors include "helping others by offering suggestions, teaching them useful knowledge or skills, directly performing some of their tasks to help out, and providing emotional support for their personal problems; cooperating with others by accepting suggestions, informing them of events they should know about, and putting team objectives ahead of personal interests; taking the initiative to do all that is necessary to accomplish objectives even if not normally a part of own duties, and finding additional productive work to perform when own duties are completed" [9, p 239].

The goal of the teammates is first to perform their tasks (i.e., taskwork, task performance) and to communicate on an as-needed basis within the context of an open space office with cubicles [10]. This communication will generate an interruption vis-à-vis the recipient, and all neighboring team members are likely to be aware of the interaction. In the context, these interruptions, which provide Team Assistances, are transgressions of an organizational norm and as such are not specifically prescribed. However, it is necessary a process ancillary to taskwork performed in a team room.

On the one hand, software engineering scientific and practitioner literatures tend to characterize ad hoc disruptions of taskwork as counterproductive interruptions and have not relied on pertinent evidence-based or theoretical models to explain or use them [11][12][13]. On the other hand, while organizational psychology has identified Team Assistances as performance-related interactions, studies in real work settings are rare, particularly regarding engineering software teams. Consequently, this study aims at gaining a better understanding of the nature, pattern, and content of Team Assistances that occur during taskwork time in software engineering teams. Second, and more specifically, we aim at examining the reasons why software developers need Team Assistance during taskwork time and what kind of knowledge is transferred during these interactions. A better understanding of these issues would provide a foundation for the study of software engineering team needs for ad hoc interactions and the speculative consequences of virtual or even absence of such ad hoc interactions on distributed software development teams. Such a study can also shed light on the appropriateness of specific practices such as occasional pair-programming [14], which can be seen as a special form of team assistance and the relevance of the use of collaborative tools for distributed team works.

In the next sections, we draw on literatures from Team Assistance to position our views in a broader context. We next describe the methods we used to analyze video recordings of software developers interactions in professional work settings. We then present our results regarding Team Assistance modality, purposes and content. These data are presented as a function of teammate roles (novice, leader, expert, developer). We conclude on the salient features of Team Assistance.

II. TEAM ASSISTANCE

Team Assistances are defined as "the discretionary provision of resources and task-related effort to another member of one's team that is intended to help that team member obtain the goals as defined by his or her role …" [15]. Essentially, Team Assistances can be seen as helping one's fellow teammates perform their role.

Team Assistances are central to the concept of adaptive team performance [17]. When one team member's task requires greater capacities than possessed, another team member can step in and compensate – the team is therefore adjusting on the spot and performing in a way not anticipated during the planning phases. These are complementary behaviors that arise either out of a specific request or merely from awareness on the part of one of the team members [17]. Unsurprisingly then, Team Assistances are a crucial form of interactions that allow a team to function as more than the sum of its individual members [18][19].

A. Team Task Characteristics

There is one caveat however. Team Assistances will arise out of a legitimate need for assistance resulting from issues with task assignment or task distribution problems [16]. The legitimacy of a need means that team members are experiencing true task difficulties beyond their capacity rather than a lack of effort [15][20]. Help provided because of social loafing or an unwarranted dependency need (when workload is in fact normal or low) is considered an illegitimate need for help and causes process loss and frictions [15][21][22]. Legitimacy of need is therefore the key situational factor that can affect the amount of Team Assistance requested or provided.

A study by Porter et al. [15] explored the personality traits of both Team Assistance recipients and providers in order to determine team composition characteristics related to the most effective use of Team Assistances. Their results showed that team members high on conscientiousness will receive more Team Assistance only when there is a legitimate need for it. These members are discriminate enough in their requests for assistance when it comes to the legitimacy of their need for it. Team members high on extraversion secured the most Team Assistances relative to members low on extraversion. There was a similar interaction effect between extraversion and legitimacy of need in terms of amount of Team Assistance received.

Porter et al. [15] also explored the personality traits most likely to lead to team members providing Team Assistances. They found that team members high on conscientiousness and emotional stability provided more Team Assistances to fellow team members, regardless of the legitimacy of need, compared to members low on these traits. Moreover, team members high on emotional stability provided even more Team Assistances if legitimacy of need was high, showing an interaction effect that the authors feel is critical to team composition. When team members are low on emotional stability, they are likely too self-focused to concentrate on the problems of fellow members, and will leave them to fend for themselves regardless of legitimacy of need.

B. Shared Mental Models

Team Assistance first and foremost requires that team members possess accurate knowledge of each other's responsibilities. Shared mental models in teams form the grounds on which team members know when to step in and provide Team Assistance, which team member should provide it, and what kind of Team Assistance is needed [16]. A team that possesses a shared mental model can anticipate and predict the needs of fellow members through a common understanding of team goals and expectations of performance. Shared models create a basic framework that promotes common understanding, as well as common action -- that is, a team that is headed toward the same goals [16]. They are particularly important in cases where a need for assistance is not initiated by a help request from the Team Assistance recipient – the need is anticipated by the Team Assistance provider, because of the shared mental model that allows predicting needs that may not be expressed [15].

Members must be willing and able to back up their fellow members – that is, they must be first aware that there is a task problem, but must also be competent in the areas of other members in order to be able to recognize when a member has problem with his/her task. The team member will have the knowledge and ability to step in and provide

compensatory behavior when a fellow member finds himself with task problems [16].

C. Team Assistance Impact on Teamwork and Team Performance

Team Assistance has direct positive effects on task performance in a team context [15], particularly in a high legitimacy condition, where the task that is causing a member to need assistance is critical to the team's performance. However, if this is not the case, Team Assistance may in fact hinder adaptive team performance by providing a behavior that is redundant [20].

The relationship between team assistances and team performance is said to be mediated by the team's ability to adapt to changes internal and external to the team (the change in environment that would lead to a workload distribution problem) [16]. Team Assistances are essential to the planning phase of teamwork, since they are demonstrative of a team's ability to adapt or revise their coordination processes if needed. This flexibility when executing teamwork plans greatly facilitates adaptive team performance in unpredictable or ever-changing contexts [17]. Teamwork is characterized by dynamic, adaptive and flexible interrelated behaviors and actions. That is, members must be able to adjust the timing of their actions and their strategy quickly in order to meet the demands of other members. This leads to – or explains the need for – coordinated and synchronized collective team action [4][19].

The preceding discussion leads to the following conclusions and research questions. Interactions during taskworks can be seen as counterproductive interruptions [23] or, as we have shown using models and empirical findings from organizational psychology, can be construed as productive Team Assistances. To explore this possibility in ongoing software development teams, we will seek to answer the following questions:

1. Are Team Assistances naturally present in software engineering teams?
2. What are the modalities of Team Assistance?
3. What are the reasons for Team Assistance?
4. What kind of knowledge is transfer during Team Assistance?

III. Field Study and Methodology

The purpose of this field study is to characterize the Team Assistance activities within a software maintenance team from an organization providing general business applications.

A. Field Study Description

This international organization has several thousand developers in many countries. In spite of the size of the organization, the setting has the attributes of smaller organizations, as development is shared among several small teams of up to 15 members each, often located at a single site. The small team observed in this field study is a stable team, whose members are used to work together and who are familiar with their tasks. There is no known conflict between the teammates and they have a respectful attitude. The four observed participants, who are all males, were part of a larger team composed of 12 individuals (1 project manager and 11 software developers) ranging widely in age, with varying levels of schooling (from a Bachelor's degree to a Ph.D. in the computer sciences and engineering), and individual experience ranging from 2 to 16 years in the field and from 9 months to 5 years of service in the company. They used a companywide software development process that is largely inspired from the waterfall model.

Physically, the participants occupied individual adjacent cubicles separated by semi-transparent walls a meter and half high. From their desks they can see whether or not their neighbors are present. Monadic (F1F) interactions occurred when participants communicate while seated at their desks. Dyadic (F2F, i.e., Face-to-Face) interactions involved two participants and they occurred when there is a movement of one of the teammates (the recipient or the provider) toward the cubicle of the other most often to gain access to an artifact. Polyadic (FnF) interactions involved more than two (n) participants and they are mostly built up from dyadic interactions. Most of the time, someone who is aware of a dyadic interaction will join his teammates to add his comment to the ongoing interaction. Dyadic and polyadic interactions required that at least one of the participants physically moved from his cubicle to another location, which was one of the other participant's cubicle most of the time.

The observed participants are described in terms of the role that each of them occupies within the team. Based on previous studies on social aspects of software engineering with this data, the roles of the four participants are the following [2][25]:

- Leader: the project manager who occupies the formal leadership position.
- Expert: the individual who is responsible for configuration management of the software built by the team, his informal leadership being rooted mostly in his knowledge and expertise.
- Developer: an individual who has no specific role on the team, formal or informal, who can be seen as the embodiment of an average developer.
- Novice: the recruit software developer who has been with this team for six months.

All procedures for these observations were approved of by independent ethics committees of both the participating organization and our University prior to the study and by each of the team members who agreed to participate in this study on a voluntary basis.

B. Recording Set-up

Video equipment was installed in the ceiling over the work area and microphones set up on various places within the working environment. Data were taken from continuous video recording during the working hours excluding lunch time. A recording session begins either in the morning or in the afternoon, and lasts half a day, with a typical duration of 2 to 3 consecutive hours. A regular session is defined as a session where all teammates are present and where there are no special events, such as meetings, visitors, etc., which

could disturb the usual task work. We retained 12 regular half-day sessions from the 23 recorded sessions. These selected sessions are evenly distributed over the two months of the recording time and account for 35 hours of video recording.

One of the researchers was a participant on the team. He was hired as a full time software developer six months prior to the study. He is identified as the Novice in this study. The purpose of being involved as a team member was to acquire the knowledge and the jargon used by the teammates in order to be able to subsequently analyze the data collected. A second researcher, who was not involved as a participant, but who had lead software development projects and has experience in video analyses, could objectively validate the observations made. Coding of the team interactions from the 12 sessions resulted in 404 Team Assistance occurrences. There was almost no e-mail exchanged between the team members, except for forwarding artifacts.

C. *Team Assistance Purposes*

We distinguish two general purposes for Team Assistance: one is to provide help to a teammate to perform his/her task and the other is to share the task with that teammate. The two purposes of Team Assistance are categorized according to the following definitions:

(1) Cooperation purpose [26] [27]: is providing feedback and coaching to increase performance. It categorizes sequences that take place when individuals provide help, but not necessarily for mutual benefit. It is characterized by informal relationships that exist without a common mission, structure, or effort. Information is shared as needed. For example, typical cooperation activities are: informal code checking, helping a teammate to set up his environment, or with a debugging task.

(2) Collaboration purpose [28]: is sharing task with a teammate. It categorizes sequences that take place when two teammates work together at an intersection of common goals, and do so by sharing knowledge, by learning, and by building consensus. This form of Team Assistance is usually an on-demand activity performed by two team members who want to work together on a specific task. Examples of collaboration are: a shared design session, and brain-storming sessions. All the collaborators have a genuine interest in the activity. We categorized only unscheduled Team Assistance collaboration sequences

These two purposes for Team Assistance can occur on various types of content, which could be related to the application or the development environment. The content of the sequences has been thoroughly studied to determine a categorization scheme for the various topics discussed. During the recording period, the team worked on 7 specific issues. To identify each of these interactions from one of these issues would make the characterization idiosyncratic and irrelevant outside this very specific field study. It was found that a more useful approach would be to define generic topics that are likely to be relevant in any software development studies. A thorough analysis of the team's project and team' interactions yielded two topic categories that were later validated successfully by the three coders.

1. Application domain related topics are associated with specific aspects or features of the software product; for example, functionality, a software component, etc. The content of the Team Assistance is based on some understanding of the application to be developed.

2. Integrated Development Environment (IDE) related topics are associated with specific aspects or features of the development environment and tools, which do not relate to the application domain, for example, programming concepts, development environment features, configuration management issues, etc.

IV. RESULTS

The results from the analyses of the 404 interaction sequences found in the 12 recorded sessions have shown that ad hoc team assistances are naturally present in collocated software engineering teams. The four observed participants spent more than a quarter (28%) of their time on cumulative Team Assistance, which in this study accounts for a total of almost 2 hours and 20 minutes per 8-hour workday per participant. The rest of the time (72%) was spent mostly on taskwork performed solo. This data support that Team Assistances are naturally present in software engineering teams. The following presents the answered resulting from this field study for each of the following three questions.

2. What are the modalities of Team Assistance?
3. What are the reasons for Team Assistance?
4. What kind of knowledge is transfer during Team Assistance?

A. *Modalities of the Team Assistances*

Fig. 1 illustrates the three modes of interaction observed during Team Assistance. All recorded Team Assistances are face to face (FtF) verbal interactions. Monadic (F1F) interaction, which we recalled, occurred when one participant communicates while seated at his desk, account for 12% of the total Team assistance occurrences. Dyadic (F2F) interactions account for 82% of all the Team Assistance recorded. Polyadic (FnF) interactions account for only 6% of all face-to-face interaction. Each of these three modes of Team Assistance behavior filled up different objectives.

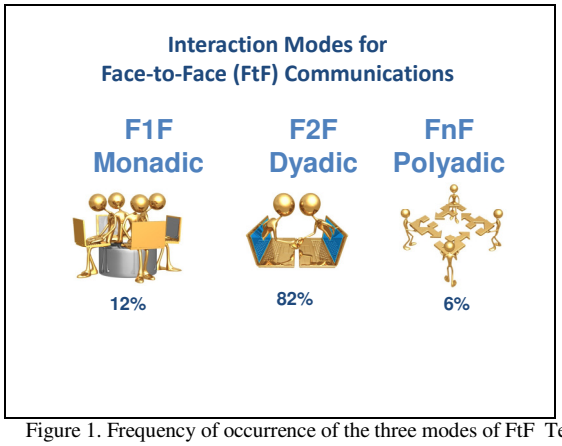


Figure 1. Frequency of occurrence of the three modes of FtF Team Assistance: Monadic (F1F), Dyadic (F2F), and Polyadic (FnF).

It has been observed in this field study that monadic interactions have short duration and contribute mostly to team awareness. For example, a teammate will state loud that he has completed the test procedure. Dyadic interactions occurred spontaneously during task work and they are truly a form of team assistance. We found that polyadic interactions last longer and are most often followed up of dyadic interactions.

Fig. 2 presents the relative frequency of the involvement of each of the four roles into each of the three modes of interaction. For example, the leader was involved in 25% of all the observed dyadic interactions (first column of the F2F mode). The Expert was involved in more than one-third of all the dyadic interactions. The total participation frequencies do not add up to 100%, because there is more than one participant for all of the interactions, except for monadic interactions (F1F).

Dyadic interaction (F2F) is the preferred mode of Team Assistance. We observe that monadic F1F interaction frequency increases when participants 'cubicle' are closer to one another. In this set up, the Developer had a central situation, he was sitting closer to the Expert than to the Leader, and the Novice was the furthest away.

B. Reasons for Team Assistances

Who are the initiators of interactions?

Does everyone initiate them occasionally, or only a few individuals do so?

Fig. 3 shows the relative frequency of interaction initiations for the four teammates who have been observed on a full-time basis.

The novice (27%) and the expert (34%) are the more frequent initiators of interactions but for different reasons. The novice was recruited on the team to add resources on various tasks but also because he had good knowledge on networks and server environments. More than 60% of his involvement in Team Assistance was initiated by him to obtain help on the understanding of the component functionalities while in 40% of his Team Assistance involvement he was as a provider of help on servers and network topics. The novice initiated interactions because he needed help for completing his task.

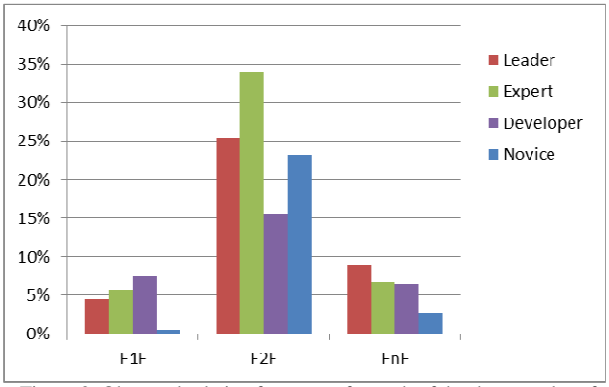


Figure 2. Observed relative frequency for each of the three modes of interaction for each role.

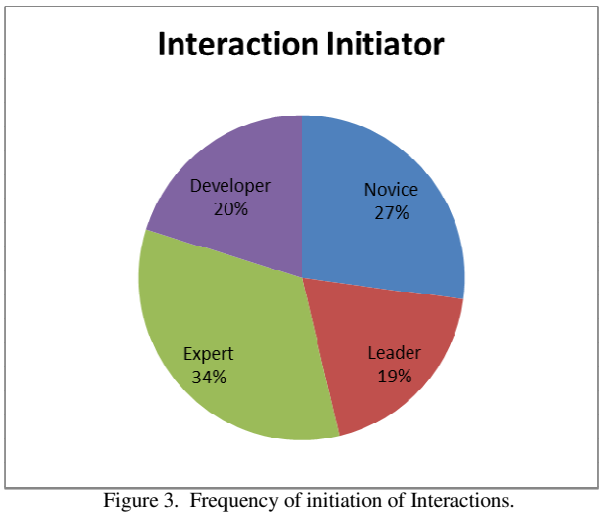


Figure 3. Frequency of initiation of Interactions.

In this context, it may sound surprising that the Expert initiated most of the requests for Team Assistance. The initiator of the interaction is not necessarily the candidate that needs Team Assistance as one could expect. A detailed analysis of the Expert interaction initiations revealed that the Expert was initiating some of the interactions for the purpose of following up on previous requests. Two cases were frequently observed. On case occurred when the Expert initiates Team Assistance interaction to provide the help that a teammate had requested earlier when the Expert cannot interrupt his work. A second case occurred when the Expert initiates interactions to follow up on previous team assistances help that were provided. He wants to make sure that the help was useful and that the recipient can proceed with his task and if needed provides additional information. That behavior was reported by the researchers in team process, as described in the previous section on shared mental model, that the teammates that have a have a high level of shared mental model can anticipate the needs for Team Assistance [15].

The Leader initiated 50% of the Team Assistance in which he was involved. The leader initiated interaction most of the time to provide information that will help the recipient. Typical cases were changes in configuration management, shared information on requested modifications to the software components. In some cases, he initiated interactions because he needs help to understand a component or the state of progress on a task. The leader was also an experienced member of the development team and he was the provider of information in most of the interactions, which he did not initiate.

The Developer initiated only 30% of the Team Assistances in which he was involved and he was exclusively a recipient concerned by technical subject related to his task. A quarter (25%) of the Team Assistances in which he was involved, as provider, had been initiated by the Expert as followed up.

C. Kind of Knowledge in Team Assistances

Fig. 4 shows the cumulative relative duration for each category of topics for collaboration and cooperation

purposes. For example, it shows that more than 60% of the time spent was on collaborative Team Assistances (see left column in Fig. 4). Most of the collaborative Team Assistance (46%) is required to solve problems related to the application domain. Since the two teammates are working toward the same goal, these Team Assistance activities contribute to shared mental model. It is observed that cooperative Team Assistances, which account in this study for almost 40% of all Team Assistance activities (see right column in Fig. 4), are mainly required to solve IDE problems and very little, less than 10% of cooperative Team Assistance activities, are undertaken to solve application related problems. We recall that all the Team Assistance activities account for almost 30% of the total time the team spent in the team room and all participants are involved at almost the same level (see Fig. 3) but in different ways as explained in the previous section.

V. DISCUSSION

This section discusses the outcomes of this observational study and shows how these outcomes can help understand the mechanics of team assistance. The threats to validity and reliability of such study are also discussed.

A. *Summary of Results*

A first observation is that all team members are almost equally involved in all three interaction modes. We found that ad hoc Team Assistance is a natural phenomenon that required almost the third of the time spent by the team members during their solo taskwork. These behaviors consist mostly of dyadic face-to-face interactions where one of the team members will visit a teammate cubicle.

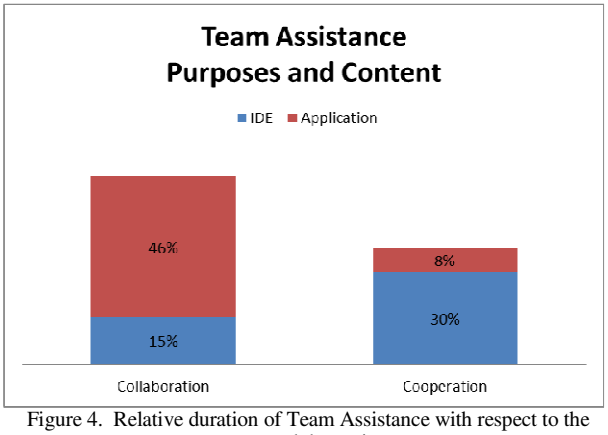


Figure 4. Relative duration of Team Assistance with respect to the purposes and the topics.

The expert role initiates almost the third of the Team Assistance interactions followed by the novice role. Although their reasons for initiating Team Assistance are different. The Novice initiated Team Assistance to obtain help on various tasks while the Expert initiates Team Assistance to follow up on requested assistance by teammates. It is noteworthy that the Leader is the one who initiated the less Team Assistance interactions.

Team Assistances are initiated for two purposes: collaboration or cooperation. Collaborative Team Assistances involve teammates sharing the same objective in assuming their tasks. Collaboration occurred mostly for increasing understanding of the application for the two teammates involved. Collaboration occurred in IDE context when developers worked together to install a server feature, for example.

Cooperative Team Assistances require that the provider teammate helps the recipient on subjects that are not immediately in line with the provider interest or task. Cooperation occurred mostly in the IDE context when teammates needed help with the configuration management systems or the debugger, for example.

B. *The Mechanics of Team Assistances*

Our observations point out to Team Assistance as an opportunistic behavior used by all participants in a co-located team. We found that each of the communication modes has a distinct purpose.

The monadic mode (F1F) contributes to maintaining team awareness. Team awareness involves knowing what activities teammates are working on and how they relate to individuals' own tasks. It allows teammates to informally communicate and coordinate their work. Burke et al. [17] explain that teams adapt to the extent that they assess the situation, formulate a plan, execute the plan, and learn from this process. In line with media richness theory [29] co-location affords teammates more opportunity for cue recognition and higher quality meaning ascription. In distributed, as well as in collocated teams, the monadic mode (F1F) can be easily computer-mediated by providing a kind of instant messaging system, where each teammate can post information judged to be valuable to maintain team awareness. The advantages of computer-mediated F1F are to avoid the interruptions caused by someone talking aloud and probably more important is the possibility of keeping track of all the messages sent.

The dyadic mode (F2F), which occurs when one teammate moves from his cubicle to communicate verbally with another, may contribute to team efficiency via what Borman [9] described as helping behaviors (i.e., citizenship behaviors). It is an opportunistic, just-in-time interaction initiated by a recipient teammate who needs information to continue his task or by a provider teammate who wants to validate help that was provided before as in the case of the Expert in this study. The degree of team efficiency, where one individual receives help and the other, who is the provider, is being interrupted, depends on the impact of the interruption on the provider. In a team room, a physical or a numerical device, such as a flag, can be raised to indicate that someone does not want to be interrupted momentarily.

The technical e-forum is a kind of asynchronous virtual F2F. A developer asks a question on the Forum, expecting that someone will answer it. The efficiency of the team room derives from the fact that the ad hoc communication is synchronous (the answer is immediate), and it involves trusted and aware co-workers. It has been observed in this

study that teammates will always prefer F2F to e-mail, within a team room.

There is a great deal of research on the difficulties involved in computer-mediating F2F communications. A verbal dialog not only allows participants to assess their understanding, but also to develop a sense of community with teammates. Most studies comparing F2F and computer-mediated communications are related to the educational environment (for tutoring) or the planned meetings. These findings cannot be readily applies to opportunistic ad hoc F2F interactions. These interactions are usually very short, and based on team awareness and the role that each teammate plays in the project. More observational and experimental studies are needed to evaluate the effectiveness of computer-mediating Team Assistance on the form of ad hoc F2F interactions, which is still the major feature of the collocated team.

The polyadic mode FnF, which occurs when an ad hoc Team Assistance involves many teammates, seems to contribute to the solution of environmental or application problems. This mode is often initiated from the dyadic mode when some issues cannot be readily resolved. When this happens, other team members may become involved and take the ad hoc Team Assistance into polyadic mode (FnF). We believe that when polyadic Team Assistance mode occurred the participants should schedule a meeting in a closed room, with only those participating who can contribute to the solution.

It is observed from this study that Team Assistance can be categorized from two purposes: cooperation or collaboration, which has been identified in independent studies [16]. Cooperation is characterized by providing help to the recipient for his own benefit, while collaboration is sharing the problem-solving task for mutual benefit. To increase the generalizability of these observations we consider the content in terms of information related to the application (like business rules) or to the development environment (IDE). It is observed that most of the collaboration occurred to increase mutual benefits of application understanding and most of the cooperation occurred to help teammates with their environment development. In terms of duration there are almost as much time spent on application understanding as on help on using the development environment.

C. Threats to Validity

Reliability and validity of the coding were assessed based on observations made on the 404 Team Assistance sequences extracted from the 12 recorded working sessions. The first step involved an intra-coder agreement, where a number of encoded data sequences were re-encoded a month later by the same coder. The second step involved an inter-coder agreement, where another coder who was able to understand the context and the jargon employed by the participants performed the same operation. Finally, the third step involved an extra-coder agreement, where an experienced coder who was not familiar with the team's work performed the same operation. An index proposed by Perreault and Leigh [24] was used to measure reliability. The inter-coder agreement indices obtained show a value of 0.89 between the two coders familiar with the team dynamics, enabling us to deduce a strong agreement. The indices obtained with the extra-coder agreement show a value of 0.72. These values suggest acceptable reliability of the coding and validity of the coding scheme. To avoid capturing behaviors that might be affected by workers' reaction to the recording equipment, interactions occurring in the first 4 weeks of the equipment's installation were not coded. Furthermore, interactions that were outside the range of cameras or microphones were deleted from the data set (n; 137).

VI. CONCLUSION

These observations confirm that Team Assistance is a core activity within a team dynamic, which may contribute to jell the team by increasing awareness, shared mental model and exchanges between the teammates. The nature of Team Assistance practices is complex and depends on various factors such as the role of the participants, the character of the individual, the physical location within the team set up, the taskwork, and the purpose and the content of the needed help.

Spontaneous interactions between collocated software developers may be perceived by practitioners and managers as undesirable interruptions that distract the developers from their tasks. However, we have observed that although it may be perceived as counterproductive interruptions, it is nevertheless a necessary – even naturally occurring – workplace behavior.

The following points are stressed based on our observational study of team dynamic:

- Interactions are legitimate, opportunistic, and of short duration;
- Almost 30% of the total team activity is devoted to spontaneous and just-in-time Team Assistance interactions
- All team members are involved in these interactions as recipients or as providers.
- Team Assistances are mostly collaborative for application domain and cooperative for development environment problems.

Our results show that Team Assistances occur without prescription from the team leader (i.e., they are ad hoc) and are an efficient means of just-in-time learning and adaptation in the workplace. It enables the initiator of the interaction to obtain quick access to information and then proceed with the task at hand. Although we did not frame our observational scheme in terms of longitudinal team development, Kozlowski et al.'s model [30] of team compilation would be an appropriate conceptual footing to examine this aspect in a future study. The compilation model argues that interactions progress from to dyadic to polyadic as people understand their respective tasks and roles.

This study was not an experiment where all the various parameters could be controlled. However, our analyses stem from reliable coding of multiple interactions that occurred over many weeks in a real-world working context. Although this study requires replication, our theoretical background and results are compelling. Our results suggest three

practices to facilitate Team Assistance and that are likely to improve team dynamics and the success of the project.

First, developers use Team Assistance for 30% of their time meaning that they do not necessarily need help for the other 70% of the time. Team leaders would be better off ensuring that Team Assistances are encouraged as long as they represent a legitimate need, and work to understand and correct non-legitimate demands.

Second, Team Assistances are sought from a pool of providers, which indicates a choice of the best provider is made. Team leaders should make sure all team members understand where each other's talents rest so that legitimate help is sought efficiently from a competent provider (e.g., application information versus IDEs).

Third, Team Assistances can be collaborative or cooperative. While both purposes foster a shared mental model, they impact different aspects of software development. Team leaders would benefit from ensuring that the correct Team Assistances are used with the appropriate task requirement from the software development life cycle.

Our results show promising avenues for future studies. One avenue would be to document Team Assistances across more teams and more project phases. This would potentially underscore how context changes the nature and frequency of Team Assistances. A second avenue would be to test whether coaching from the team leader can help leverage the impact of Team Assistances. Hackman and Wageman [31] suggest a theory of team coaching that hinges on three components one of which is consultation on team processes. A leader that consults his/her team mid-way within a project phase is likely to identify if and how team members engage in Team Assistances. The theory predicts that such a consultation is likely to foster more efficient team process. A third avenue would be to measure performance such that the efficacy of Team Assistances can be assessed against mainstays of team performance such as proficiency, adaptability and pro-activity [32].

ACKNOWLEDGMENT

This research would not have been possible without the agreement of the company in which it was conducted, and without the generous participation and patience of the software development team members from whom the data were collected. To all these people, we extend our grateful thanks. This research was supported in part by NSERC grant A-0141.

REFERENCES

[1] P. N. Robillard and S. Cherry, "Types of knowledge exchange during team interactions: A software engineering study," The sixth International Conference on Information, Process and Knowledge Management, eKNOW 2014, pp. 131-136.

[2] S. Cherry and P. N. Robillard, "The social side of software engineering – A real ad hoc collaboration network," Int. J. Human-Computer Studies (IJHCS), Vol. 66, 2008, pp. 495-505, doi:10.1016/j.ihcs.2008.01.02.

[3] M. A. Marks, J. E. Mathieu, and S. J. Zaccaro, "A temporally based framework and taxonomy of team processes," Academy of Management Review, 26, 2001, pp. 356-376.

[4] F. Chiocchio, S. Grenier, T. O'Neill, K. Savaria, and D. J. Willms, "The effects of collaboration on performance: A multilevel validation in project teams," International Journal of Project Organisation and Management, 4, 2012, pp. 1-37.

[5] P. D'Astous and P. N. Robillard, "Empirical study of exchange patterns during software peer review meetings," Information and Software Technology, vol. 44, no. 11, 2002, pp. 639-648.

[6] S. W. J. Kozlowski and D. R. Ilgen, "Enhancing the effectiveness of work groups and teams," Psychological Science in the Public Interest, vol. 7, no. 3, 2006, pp. 77-124.

[7] S. J. Motowidlo, Job performance. In W. C. Borman, D. R. Ilgen, R. Klimoski, and I. B. Weiner (Eds.), Handbook of Psychology: Industrial and Organizational Psychology, vol. 12, pp. 39-53. London: Wiley. 2003.

[8] D. W. Organ, "Organizational citizenship behavior: The good soldier syndrome," Lexington, MA: Lexington Books. 1988.

[9] W. C. Borman, "The concept of organizational citizenship," current directions in Psychological Science, vol. 13, no. 6, 2004, pp. 238-241, doi: 10.1111/j.0963-7214.2004.00316.x

[10] A. J. Ko, R. DeLine, and G. Venolia, "Information needs in collocated software development teams," Proceedings of the 29th International Conference on Software Engineering, 2007, pp. 344-353.

[11] E. Arroya, T. Selker, and A. Stouffs, "Interrruptions as multimodal outputs: Which are the less disruptive?" in Proceedings of the Fourth IEEE International Conference on Multimodal Interfaces (ICMI'02), 2002, 4 pages.

[12] T. D. LaToza, G. Venolia, and R. DeLine, "Maintaining mental models: A study of developer work habits," Proceedings of the 28th International Conference on Software Engineering, 2006, pp. 492-501.

[13] A. J. Ko, B. A. Myers, M. J. Coblenz, and H. H. Aung, "An exploratory study of how developers seek, relate, and collect relevant information during software maintenance tasks," IEEE Transactions on Software Engineering, vol. 32, no. 12, 2006, pp. 971-987.

[14] M. Ally, F. Darroch, and M. Toleman, "A framework for understanding the factors influencing pair programming success," in 6th International Conference on Extreme Programming and Agile Processes in Software Engineering, 2005, pp. 82–91.

[15] C. O. L. H. Porter, J. R. Hollenbeck, D. R. Ilgen, A. P. J. Ellis, B. J. West, and H. Moon, "Backing up behaviors in teams: The role of personality and legitimacy of need," Journal of Applied Psychology, vol. 88, no. 3, 2003, pp. 391-403, doi: 10.1037/0021-9010.88.3.391

[16] E. Salas, D. E. Sims, and C. S. Burke, "Is there a "Big Five" in teamwork?" Small Group Research, vol. 36, no. 5, 2005, pp. 555-599, doi: 10.1177/1046496405277134.

[17] C. S. Burke, K. C. Stagl, E. Salas, L. Pierce, and D. Kendall, "Understanding team adaptation: A conceptual analysis and model," Journal of Applied Psychology, vol. 91, no. 6, 2006, pp. 1189-1207, doi: 10.1037/0021-9010.91.6.1189

[18] R. M. McIntyre and E. Salas, "Measuring and managing for team performance: Emerging principles from complex environments," In R. A. Guzzo & E. Salas (Eds.), Team effectivness and decision making in organizations, pp. 9-45, San Francisco: Jossey Bass. 1995.

[19] E. Salas, C. S. Burke, and J. A. Cannon-Bowers, "Teamwork: emerging principles," International Journal of Management

Reviews, vol. 2, no. 4, 2000, pp. 339-356. doi: 10.1111/1468-2370.00046

[20] D. R. Ilgen, J. R. Hollenbeck, M. Johnson, and D. Jundt, "Teams in organizations: From Input Process-Output models to IMOI models," Annual Review of Psychology, vol 56, 2005, pp. 517-543, doi: 10.1146/annurev.psych. 56.091103. 070250.

[21] I. D. Steiner, Group Process and Productivity. New York: Academic Press. 1972.

[22] C. M. Barnes, J. R. Hollenbeck, D. T. Wagner, D. S. DeRue, J. D. Nahrgang, and K. M. Schwind, "Harmful help: The costs of backing-up behavior in teams," Journal of Applied Psychology, vol. 93, no. 3, 2008, pp. 529-539, doi: 10.1037/0021-9010.93.3.529

[23] Q. R. Jett, and J. M. George, "Work interrupted: A closer look at the role of interruptions in organizational life," Academy of Management Review, vol. 28, 2003, pp. 494-507.

[24] W. D. Perreault and L. E. Leigh, "Reliability of nominal data based on qualitative judgments," Journal of Marketing Research, vol. 26 May, 1989, pp. 135-148.

[25] J. Conny, P. A. V. Hall, and M. Coquard, "Talk to Paula and Peter -- They are experienced: The experience engine in a nutshell," SEKE 2000, 2000, pp. 171-185.

[26] S. M. Hord, "Working together: Cooperation or collaboration," Austin, TX: Research and Development Center for Teacher Education, University of Texas. (ERIC Document Reproduction Service No. ED 226 450), 1981.

[27] J. Roschelle, and S. D. Teasley, "The construction of shared knowledge in collaborative problem solving," In Computer Supported Collaborative Learning, C. E. O'Malley, ed. Springer-Verlag, 1995, pp. 69-97.

[28] P. Dillenbourg, "What do you mean by collaborative learning?" In P. Dillenbourg (Ed) Collaborative-learning: Cognitive and Computational Approaches, Elsevier, Oxford, 1999, pp. 1-19.

[29] R. L. Daft and R. H. Lengel, "Information richness: A new approach to managerial behavior and organization design," Research in Organizational Behavior, vol. 6, 1984, pp. 191-233.

[30] S. W. J. Kozlowski and K. J. Klein, "A multilevel approach to theory and research in organizations: Contextual, temporal, and emergent processes," In K. K. Klein & S. W. J. Kozlowski (Eds.), Multilevel theory, research, and methods in organizations, pp. 3-90. San Francisco: Jossey-Bass. 2000.

[31] J. R. Hackman and R. Wageman, "Asking the right questions about leadership," American Psychologist, vol. 62, 2007, pp.43-47.

[32] M. A Griffin, A. Neal, and S. K. Parker, "A new model of work role performance: positive behavior in uncertain and interdependent contexts," Academy of Management Journal, vol. 50, no. 2, 2007, pp. 327-47.

Development and Evaluation of CSCL System for Large Classrooms Using Question-Posing Script

Taketoshi Inaba and Kimihiko Ando
Graduate School of Bionics, Computer and Media Sciences
Tokyo University of Technology
Tokyo, Japan
e-mail: inaba@stf.teu.ac.jp, e-mail: ando@stf.teu.ac.jp

***Abstract*—In the area of Computer Supported Collaborative Learning (CSCL) research, scripting collaborative learning is a relatively new but promising approach to promote learning. The term scripting is used to describe ways of prescribing relevant elements for collaborative interaction, such as group formation, roles, learning activities, sequence of learning activities. Many studies have shown that free collaboration without explicit scaffolding rarely produces effective interaction and that the script can be one of the most effective scaffoldings. Basing on SWISH model proposed by Dillenbourg, we have adopted the reciprocal teaching approach and designed a script which allows students to create questions and answer them mutually. To implement this question-posing script for large classrooms, we have developed a CSCL system which has two important functions: automated group formation function that can form groups on the fly, based on students' personal traits, and chat function by which students can discuss each other within their group. For the evaluation, we have conducted an experiment with some 300 students in a large classroom to evaluate our system and analyze interactions in detail during each sequence of learning activities. The evaluation result indicates that the learners felt encouraged to understand better about learning task. At the same time, it becomes clear that the quality of discussion on chat affects reciprocal question posing. As well, it is indicated that group size and knowledge level of leader or other members affect the process of reciprocal actions and activities at some degree.**

Keywords-Collaborative learning; CSCL; large classroom; collaborative script; question-posing

I. INTRODUCTION

This article is an extended version of a conference paper presented at eLmL 2014, the Sixth International Conference on Mobile, Hybrid and On-line Learning [1]. It introduces more information on the theoretical background of this study, a more specific and technical presentation of the system and some new data from the experiment.

A. CSCL and its issues

According to the social constructionism presented by Vygotsky [2] and the theory of legitimate peripheral participation presented by Lave and Wenger [3], the learning, which was understood as a cognitive process in an interior of an individual learner, will be recognized as a social process, or social cognition that progresses while cooperating with others [4]. Far from denying the learning as an individual cognitive activity, the social cognition can promote knowledge construction at an individual level and metacognition for learning strategies, through problem-solving by discussing with others [5].

The environment for such collaborative learning is built on the computer network, and such computer technologies are used as a supporting tool to promote collaborative learning, which is called, Computer Supported Collaborative Learning (CSCL). Advantages of CSCL over the face-to-face learning are: learners who are geographically or timely distant from each other can learn, a large number of learners can learn and be managed, logs of the learning process in details can be saved for learners, managers and scholars to re-use them, learning software and contents can be used and many more.

On the other hand, many case studies on the collaborative learning point out that it is highly unlikely for learners to carry out collaborative activities voluntarily while learning without an external scaffolding [6] [7]. For this reason, in order to resolve such issues in learning, various methods have been developed to appropriately regulate and structure the learning process within a group for effective and productive work and discussions among learners.

In this study, one of such methods, "collaborative script" was implemented in the CSCL system and used in a large classroom in the university. First, the next section will provide the overview of the collaborative script.

B. Collaborative script and its issues

The concept of script was originally suggested by Schank and Abelson in the field of cognitive science, and it has a meaning of internalized knowledge about socially sharing steps and rules people should follow in a certain situation (e.g., eating at a restaurant) [8].

Once the concept was introduced in the field of collaborative study, the script became a series of external scaffolding methods that are provided to promote collaborative learning. The first study on collaborative script was proposed by O'Donell and Dansereau [9] [10], which defines the script as a scenario for a small learning group,

which prescribes in details, who is carrying out what kind of learning activities and when. Due to the complexity of the script before the learning activities themselves, learners needed to be trained to follow the script.

After the script was adopted in CSCL, instead of training learners to execute the script prior to learning, the system interface was used to indirectly lead them to the scripted learning process [11].

Many researches indicate that the script can be designed at 2 levels in the CSCL environment. First, there is a design approach at a macro level; it defines who will learn, what assignment subjects for a group and how to distribute tasks among learners. On the other hand, there is a micro level approach which consists in prescribing the details of each learning activity in order to revitalize social interactions among learners [12] [13].

There have been many studies that indicate the effectiveness of various CSCL systems with the script, but there are some issues at the same time. First, there is an issue on controlling a compelling power of the script. In other words, it means how to deal with the risk of over-scripting which takes too much self-motivation out from learners [14]. Next, despite a lot of empirical case studies, yet there are very few suggestion on a script design model that can be commonly used, with some exceptions [15] [16] [17]. About the first issue, we suggested previously a method to flexibly adjust compelling power of the script according to learners' traits and learning situation [18]. So, this study focuses on the second issue, adopting a design method as the approach in order to design the script based on the design principle and implement and assess it.

C. SWISH MODEL as Design Principle

The purpose of the collaborative script is to support the problem solving and knowledge construction by social interactions among learners. To do so, a mechanism to trigger effective interactions is an important element. A Swiss scholar, Dillenbourg, suggests SWISH model as such mechanism. This model is the design principle for collaborative script that gives tasks that would generate conflicts among learners; it is supposed to promote intense interactions (statements, explanations, discussion, etc.) to overcome these conflicts [13].

Exactly, SWISH is an abbreviation of "Split Where Interaction Should Happen". And this model can be formulated in three points:

1. Learning results from the interactions while students are constructing a shared understanding of the task despite the fact that the task is distributed.
2. Task distribution determines the nature of interactions. Interactions are mechanisms for overcoming task splits.
3. Task splits can be designed for triggering the interactions that designer wants to elicit.

From this model, three script schemata are drawn as design guidelines: 1. jigsaw schema, 2. conflict schema, 3. reciprocal schema. In the jigsaw schema, the information necessary to solve the problem being distributed, no group member is able to solve the problem alone. This split elicits social interactions to seek mutually the solutions in bringing complementary knowledge each other. The conflict model forms groups with students having conflicting opinions; this conflicting relation elicits argumentation.

In this study, we adopt the third schema, reciprocal one. This schema defines the roles for each student and switches these roles. The horizontal split is realized between cognitive and metacognitive layers of the task and is counterbalanced by reciprocal regulation. The most well-known example of this schema is Palinsca and Brown's reciprocal teaching method [19]. In their approach for enhancing reading skill, four roles (questioner, summarizer, clarifier, predictor) are assumed in rotation by students. Through the reciprocal teaching process, the activation of mutual monitoring activity is particularly expected; learning accuracy is monitored during asking questions or clarifying and summarizing the content, whereas learning consistency of predictions is assessed.

According to Dillenbourg, by using collaborative script, the entire learning process is composed of multiple phases that are linear occurrence in succession [14]. Each phase has attributes, being regulated by: 1. Type of task, 2. Group structure, 3. Tasks assigned to group members, 4. Communication method and 5. Required time. As it will be shown in Section II, in conformity with the above, our script proposed in this study can be outlined as follows: 1. Tasks for the major phase is to prepare questions and discuss/refine the questions reciprocally, 2. The groups have 3 to 5 members (depending on the system specifications, a number of group members can be flexible) 3. Tasks are assigned to question preparer, answerer and grader based on reciprocal tutoring method, 4. The major communication method is to chat, using the network and 5. Time required is a deadline for the final project to be submitted, which is the end of the class.

Also, many existing systems have a control function in place such as an order in making comments and attributes of comments (suggestion, question, approval, disapproval, etc.) [20] [21]. This study, on the other hand, does not have such control in place at this time. We felt that such function to control attributes and occurrence of comments is unnecessary when the conditions are narrow and limited such as to prepare questions and allocating tasks to each leaner.

D. Structure of this paper

This paper is structured as follows. Section II presents the general outline and the purpose of this study, and Section III describes our CSCL system for large classrooms. The collaborative script design is discussed in Section IV. In the Section V, the details of page structure is described with their function. Then, we present our experiment and results from our evaluation in Sections VI and VII. Section VIII concludes the paper.

II. PURPOSE OF THE STUDY

In this study, the script based on the reciprocal schema, is designed and implemented in the system to assess its effects. The system is for an environment where several hundred students in higher educational institutions cannot interact

with one another face-to-face. The collaborative learning is carried out by those students using the system online.

As for the assessment, assignments and chat log data are used to assess the quality of interactions during the collaborative process and its learning effects. By analyzing the correlativity between the two, we aim to have some guidelines for improving the script and design principle.

III. SYSTEM

As Fig. 1 shows, our system was developed for an environment, such as a large classroom with several hundred people at higher educational institutions where face-to-face group learning is difficult. A teacher and students gain access to the CSCL server through PCs that are connected to the network. Learners can form a group regardless of where their locations are, and a teacher can remotely keep track of learning state of each group.

Our system is a server-client web application. As Fig. 2 shows, Linux server was constructed by using Java. We used Apache for Web server and Tomcat for Web container. The application was realized by JSP and servlet. Mysql was used for the data base in which information about the script and users properties is contained.

On client-side, there is, practically, no limitation about the choice of OS and browsers, but the use of Windows is recommended

A. System Overview

As Fig. 3 shows, the system consists of different functions, such as "automated group formation" and "questionnaire preparation" by which a teacher designs a collaborative learning, "assignment submission", "reciprocal reviews" and "chat within a group" that provide a collaborative environment to learners. "Learners' properties" in Fig. 3 are drawn from questionnaires and pre-tests that were administrated before. Based on the properties, the system automatically forms groups.

B. Flow of Collaborative Learning

The collaborative learning in this system is composed of 5 blocks, as Fig. 4 shows. The following is the learning flow.

1. "Prior Setting" allows a teacher to conduct questionnaires, prepare pre-tests and register to the system.
2. In "Pre-learning", each learner submits the questionnaire and pre-test, which was registered in "Prior Setting" on the system.
3. In "Group Formation", the system automatically forms groups based on the parameters the teacher has set and results of statements/answers by the learners. Small adjustments to the group formation can be made manually by the teacher.
4. In "Collaborative Learning", reciprocal reviews within a group and among groups as well as chat system within a group can be done in the system. The learners carry out these collaborative works according to the collaborative script.
5. In "Post Assessment", the teacher reviews and grades submitted assignments.

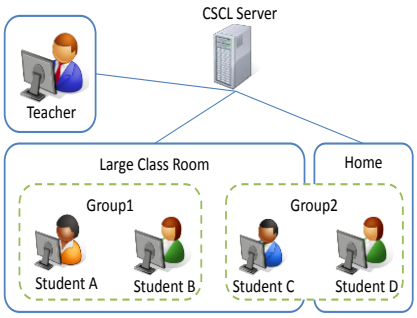


Figure 1. System overview

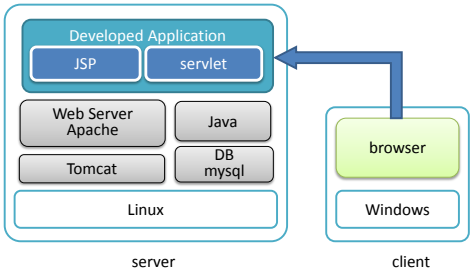


Figure 2. Technical Details

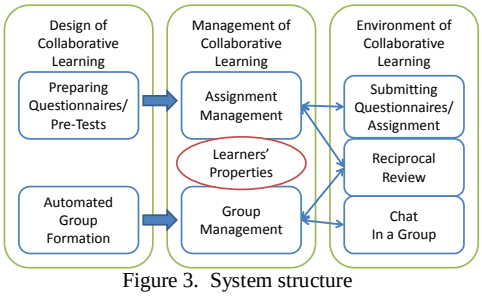


Figure 3. System structure

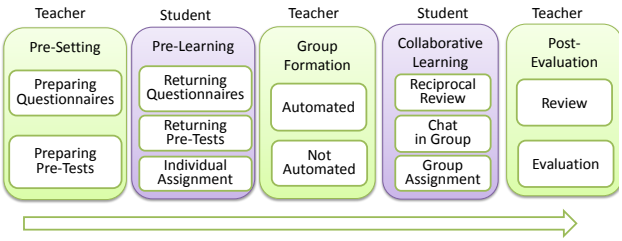


Figure 4. Flow of collaborative learning suggested by the system

C. Automated Group Formation Function

In this study, group formations are made possible in various ways that a teacher intends to do, by combining multiple elements of user characteristics that are obtained beforehand.

For example, a teacher can freely decide how many people to be in a group. He can also form flexibly groups with members of which properties are similar, or different.

Our system has two possibities for group formation; the first possibilty is to form groups with homogeneous students who have similar properties, the second is to form groups with heterogeneous students who have different properties. These properties are extracted from the test score or from the result of questionnaire, and then they are represented as numeric values .

Fig. 5 shows the case of group formation with 3 students. At first, the numeric values are sorted. For forming homogeneous groupus, three students are picked up in number order, from the first to the last (Fig. 5). In contrast, for forming heterogenous groups, each student is distributed to each group from the first student to the last student (Fig. 6).

D. Collaborative Script Function

In collaborative script, tasks are assigned according to roles, such as “Preparer”, “Answerer” and “Grader”. In the system, the group management function assigns tasks to each learner while the assignment management distributes allocated tasks. Also, roles that each learner is supposed to play and tasks are given automatically so that learners can work on their tasks at an appropriate speed without having to think about the collaborative script.

IV. COLLABORATIVE SCRIPT DESIGN

Supposing the experimental environment shown in Table I, the details of the collaborative script to be executed in the proposed system were designed.

A. Question-Posing Script

A script was made for the learning process in the task model called “reciprocal question-posing”. The following is a flow of “reciprocal question-posing collaborative script”, which was designed in this experiment.

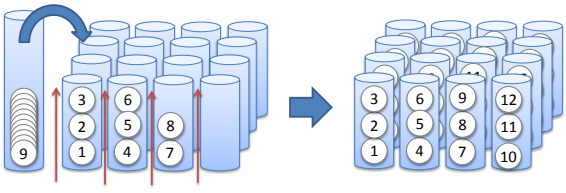


Figure 5. Formation of homogenous group

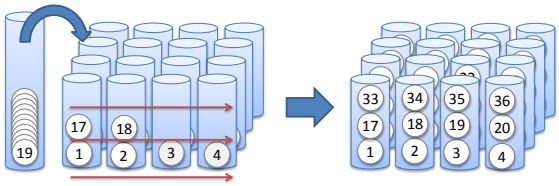


Figure 6. Formation of heterogeneous group

Phase-1: Preparing individual questions
A theme of question posing is given to learners. All the students prepare a question based on the given theme and submit it, including the answer and explanation about the question.

Phase-2: Reviews within group
Regarding the question prepared at Phase-1, 3 members within a group are assigned as a question preparer, answerer and grader and review reciprocally within the group through the following activities (Fig. 7).
a. An answerer prepares answers to the questions prepared by a question preparer and submits the answer and evaluation of the question.
b. A grader grades the answer submitted by the answerer in a. and submits the graded result and evaluation of the question.
c. Based on the evaluation submitted in a. and b. a question preparer evaluates himself/herself,
d. The above process from a to c is repeated until all the learners rotate to take a different role within the group and become a question preparer

Phase-3: Question preparation within a group
Through a discussion in a group chat, a question must be prepared for submission. The answer and explanation are prepared along with the question.

Phase-4: Submission and publish of final questions
Students submit a question/answer/explanation to their teacher. The teacher then publishes the questions as a assignment among groups.

Phase-5: Solving questions reciprocally among groups
Students solve group questions that are published.

V. PAGE STRUCTRE

In this section, the page structure of our system will be shown below with Webpage transition diagrams.

TABLE I. PRECONDITION OF COLLABORATIVE SCRIPT

Number of Students	Aboue 300 people
Member of Groups	3 people
Learning Time	90min × 2
Design Guideline	Reciprocal Teaching

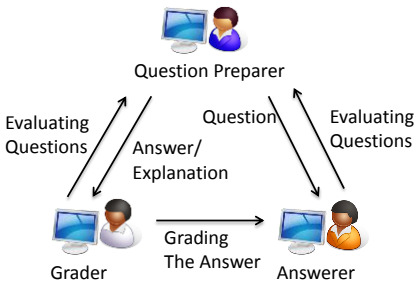


Figure 7. Group review

As Fig. 8 shows, the system has three distinct subsystems: student subsystem, teacher subsystem, administrator subsystem. Each subsystem also has its own subsystems. In the following subsections, their functions with webpage transition are described.

A. Student subsystem

The student subsystem is composed of the main system and the CSCL system. Fig. 9 is a page transition diagram of the main system. In this system for students, functions like student registration and student login are set up. In My Page after the login page, students can do course registration and respond to questionnaires. From the data collected in these pages, user model of each student is constructed for automated group formation. After these pages, students enter into the Forum Login Page which leads to the CSCL system.

Fig. 10 recapitulates the main steps by which students move from the student registration to the Forum login.

Fig. 11 shows the page transition of the CSCL subsystem after the Forum login which is opened to the students who have been assigned to a group after course registration. To execute the question-posing script explained in Section IV.*A*, this subsystem have main functions such as individual question submission, answer to question and evaluation, question grading and evaluation, question self-evaluation, group chat BBS, group assignment submission and so on.

Fig. 12 presents the flow of main student activities defined by the script. But if necessary, students can return to prior activities.

B. Teacher subsystem

The teacher system consists of the main system and the group formation system.

Fig. 13 is a page transition diagram of the main system which has basic functions like teacher registration and teacher login. In My Page after the login, teachers can registrate their courses and make questionnaires. Since questionaire items are shared by all teachers, it is necessary to check the list of existing items before the new items registration.

Fig. 14 shows the page transition of the group formation subsystem: teachers have roughly two possibilities in forming groups. The first possibility is to select questionnaire items and form groups on the basis of their result. The second possibility is to form groups from the result of test scores.

C. Administrator subsystem

The main system is the singular component of the administrator subsystem. Fig. 15 shows the page transition of this component. In the questionnaire classification registration, the administrator can determine what kind of subject (favorite subject, learning style, preferences, characters etc.) the questionnaire is addressing. In the questionnaire type registration, he can define the type of questionnaire (free writing, fill-in-the-blank, multiple-

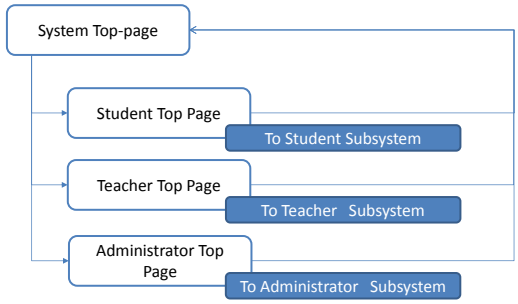


Figure 8. Webpage Transition Diagram

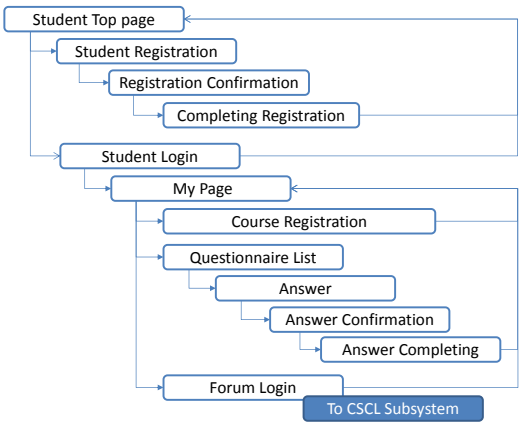


Figure 9. Student Subsystem

Figure 10. Steps of student activity

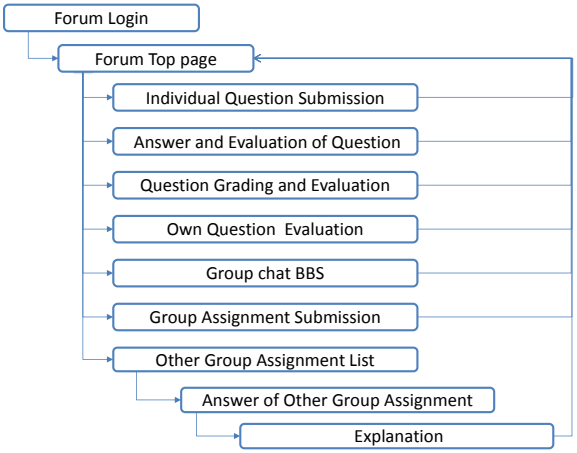


Figure 11. CSCL Subsystem

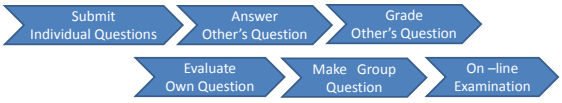


Figure 12. Steps of student activity in the CSCL Subsystem

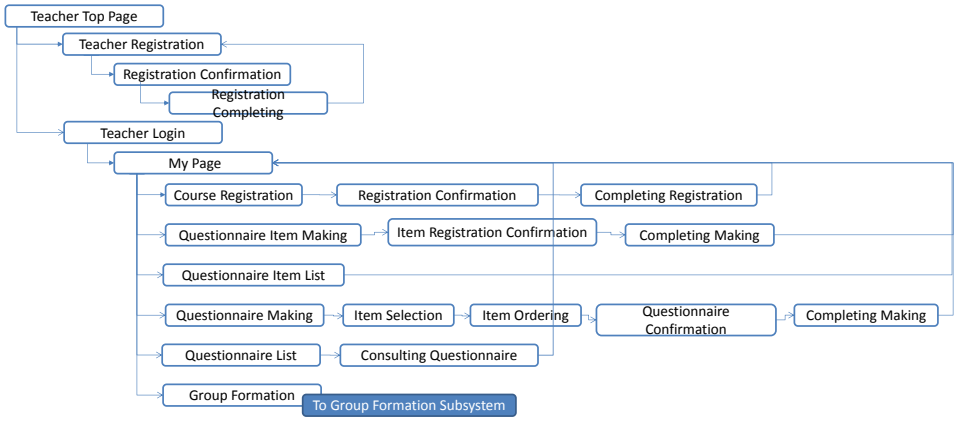


Figure 13. Teacher Subsystem

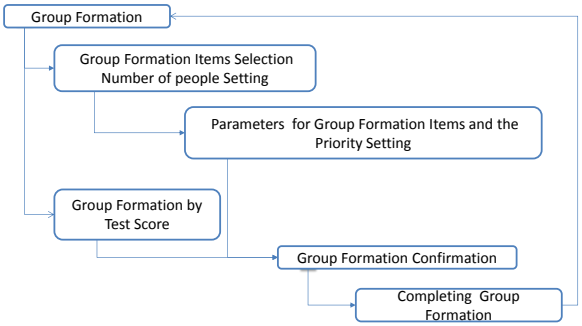


Figure 14. Group Formation Sub System

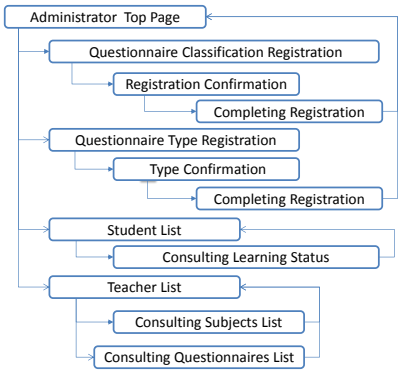


Figure 15. Administrator's Subsystem

choice, etc.). He can also consult the student list, the actual learning status of each student, the teacher list and all the data of questionnaires.

VI. EXPERIMENT OVERVIEW

To assess this system, an experiment was carried out during a class at Tokyo University of Technology. The overview is as follows:

・Targets: Students at Tokyo University of Technology Freshman to Senior 298 students, 112 groups

・Dates for the experiment: January 10 (Tue) and January 18 (Wed), 2011

・Lecture: Basics of the logic

・Learning assignment: students prepare a question; the question has statements in Japanese that represent an deductive inference that contain several premises and a conclusion. The answer must have a well–formed formula that represents correctly the inference, and a truth table that verifies the validity/invalidity of the inference. For this assignment, several exercises had been done during previous lectures. Also, similar question were distributed and completed as a pre-test one week before the experiment. The pre-test was graded by the teacher in charge.

The experiment was carried out during 2 days in a 90 minute class. On day 1, 60 minutes were spent for answering/evaluating reciprocally within each group. On day 2, another 60 minutes were spent for posing questions reciprocally within each group. The flows for learning are shown in Fig. 16.

The group review phase for day 1 is for answering/evaluating questions, grading/evaluating questions and self-evaluation. Fig. 17 shows evaluations of a question by a grader's point of view.

The group review phase for Day 2 is for preparing group question. Using a group chat function, learners discuss how to pose the final question.

In this experiment, a number of group members was set to 3. But there were some groups of less than 3 group members due to no attendance of some members. Specially, since groups could not be changed on Day 1 and Day 2, there were many groups of less than 3 group members due to no attendance of group members on Day 2. For this reason, the evaluation of this experiment was done on only 93 groups with group members of 2 or 3 on Day 2. Table II shows changes in a number of group members.

Also, on Day 1 carry out a group review, group members of less than 2 members could not carry out a group review. In this case, the groups of 2 members continued the learning using a different script that allows the 2 members

solved questions and graded reciprocally. For a group of 1 member, the 1 member had additional members who came in late.

VII. EVALUATION

The aim of this section is to present the results of the experiment and their evaluation in different ways.

A. Automated Group Formation

In this experiment, groups were formed in a way that the academic level for each group is similar. Each group consists of equal numbers of learners who ranked top, middle and low in the pre-tests about the content of the lecture. The results of the pre-tests were total points (perfect score is 400 points) of 4 pre-tests that had been implemented according to the progress of the lecture. All the grading was done by the same teacher. Fig. 18 shows the distribution of individual score and average score within group. Because the average scores gather in the median, the automated group formation functions normally.

B. Question-Posing Script Evaluated by Learners

At the end of the experiment, we distributed a questionnaire to the students. Fig. 19 shows the responses to the question "Did you have a deeper understanding through posing questions?" Since many responded, "Deepened" and few answered, "Not deepened" and "Not at all deepened", the learners find the script effective.

Fig. 20 shows the degree of difficulty in posing questions. "Very difficult" (18%) and "Difficult" (68%) form a large majority. This result indicates the high degree of difficulty for students while posing questions. And between the degree of understanding deepness and the degree of difficulty, there is a very strong correlation (r=0.98), which shows a trend that the higher is the difficulty, the deeper is the understanding.

Fig. 21 shows the degree of interest in posing questions. Almost half of responses are positive ones ("Very interesting" and "Interesting"). Between the degree of interest and the degree of understanding deepness, there is a strong correlation (r=0.82), which shows a trend that the more interesting is the question-posing the deeper is the understanding.

Fig. 22 shows the responses to the question, "what was

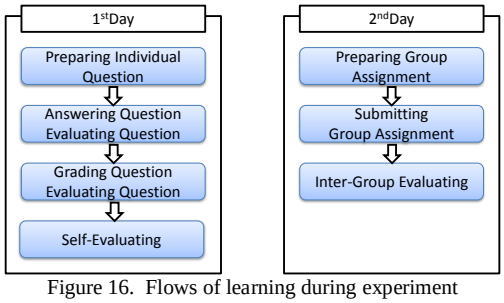
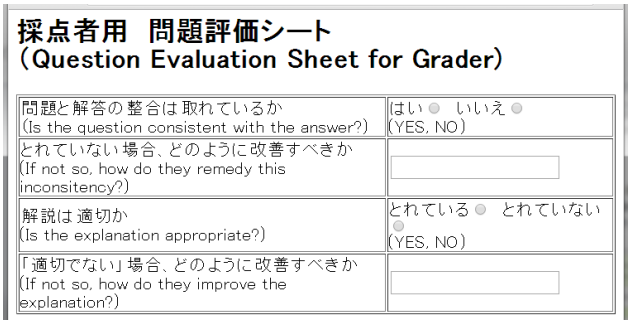


Figure 16. Flows of learning during experiment


採点者用　問題評価シート
(Question Evaluation Sheet for Grader)

問題と解答の整合は取れているか (Is the question consistent with the answer?)	はい ○ いいえ ○ (YES, NO)
とれていない場合、どのように改善すべきか (If not so, how do they remedy this inconsistency?)	
解説は適切か (Is the explanation appropriate?)	とれている ○ とれていない ○ (YES, NO)
「適切でない」場合、どのように改善すべきか (If not so, how do they improve the explanation?)	

Figure 17. Evaluations of a question by a grader's point of view

TABLE II. CHANGES IN A NUMBER OF GROUP MEMBERS

Number of Members	Number of Groups	
	1st Day	2nd Day
3	77	40
2	32	53
1	3	15

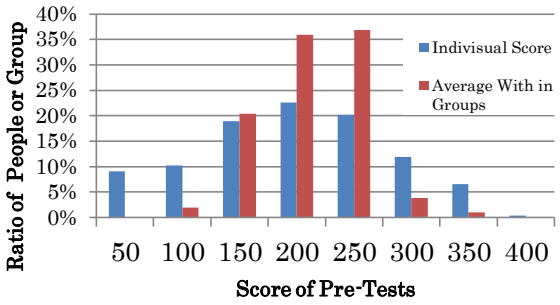


Figure 18. Distribution of individual score and average score within groups

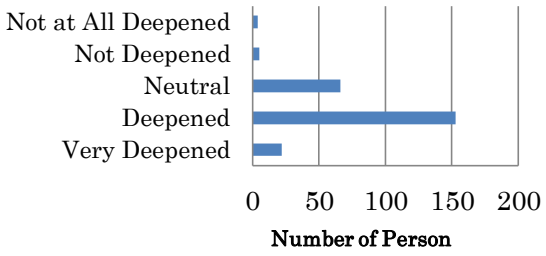


Figure 19. Responses to the question "Did you have a deeper understanding through posing questions?"

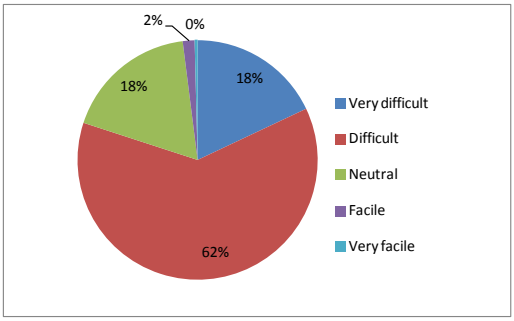


Figure 20. Responses to the question "Was it difficult to pose questions

the most useful reference while question-posing?". Responses as "Chat within Group", "Evaluation on questions by answerers" and "Evaluation on questions by a grader", of which teamwork take a large part, were highly evaluated.

C. Interaction within Groups

Contents of the chat were divided up into the following 5 categories: "Detailed discussion on important points", "Discussion on important points", "Discussion that often went off on a tangent", "Discussion that were mostly chit-chatting" and "Pointless discussion". The categories are shown in Table III. We fixed these categories after the attentive reading of the contents of the chat. The evaluation was executed by 1 person according to the evaluation standard while the other checked the result.

Tables IV to VI are extracted from the chat logs. Table IV shows a part of discussions that was evaluated as "Detailed discussion on important points". It shows that 3 people consulted with one another on how to carry on.

Table V shows a part of discussions that was evaluated as "Discussion on important points". It shows that only some casual conversations were the basis for making a decision to carry on. Even after the conversations, there were many communications to inform what had been decided and agreements on what had been decided. "Going off on a tangent" contained chit-chatting in the above conversations while "More chit-chatting" had more chit-chatting than discussions.

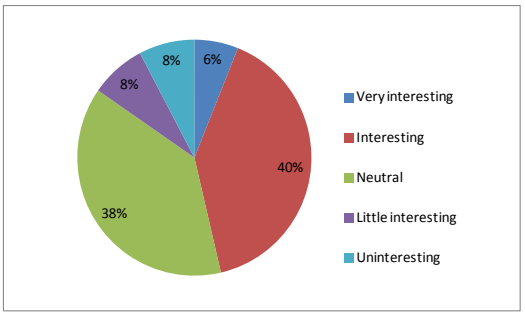


Figure 21. Responses to the question "Was it interesting to pose questions

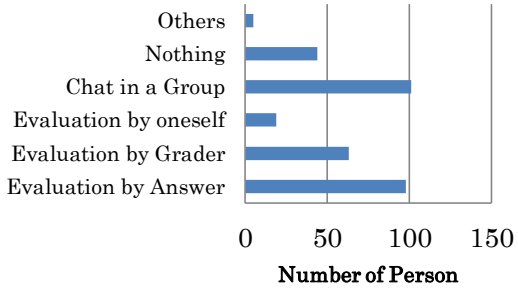


Figure 22. Responses to the question "What was the most useful reference while question-posing?"

Table VI shows a part of discussions that was evaluated as "Pointless". It shows that the conversations were going into a direction of avoiding deep discussions.

Fig. 23 shows the quality of discussions by each group, of which chat logs were evaluated. In both groups of 2 or 3 people, more than 70% of all the groups fell into either one of the 2 categories, "Detailed discussion on important points" and "Detailed discussion", meaning that many groups had good interactions.

Fig. 24 shows the number of statements made per person within each group. In the groups of 2 people, an average number of statements made per person is 26.2 while in the groups of 3 people, the average was 22.3. These results suggest that in both groups, relatively active discussions were held, and the interactions were sufficiently activated. Also, a number of statements was higher in the groups of 2 people rather than in the groups of 3.

Fig. 25 shows the comparison between the average scores of the pre-tests within each group and the qualities of the discussions. When the average scores were divided into

TABLE III. QUALITY OF DISCUSSION

Detailed Discussion on Importnant Points	Participants discuss carefully and meticulously to decide how to carry on.
Discussion on Important Points	Decision are taken by short discussions. Assignments are completed rapdely with modifications.
Often Went Off on a Tangent	Participants discuss on important points. But they chitchat often.
Mostly Chit-Chatting	Participants chitchat more often.
Pointless Discussion	Participants always chitchat and don't try to complete the assignments

TABLE IV. EXAMPLES OF "DETAILED DISCUSSION ON IMPORTANT POINTS"

Talker	Contents
D	Where do you want to change?
E	That's right … I guess, first of all, we definitely need to change the question, and then, what about the well-formed formula?
D	How is it that changes only the third line of the question?
D	Regarding the well-formed formula, it's the final part after ⊃.
E	That's good idea.
F	I agree. How do we want to change that?

TABLE V. EXAPMPLE OF "DISCUSSSION ON IMPORTANT POINTS"

Talker	Contents
G	Whose problem will we use?
H	How about I's Question? I don't have any particular reason for it though.
I	I think it's OK if it's corrected.
H	Then, let's make corrections on I's question and use itI.
G	All right, let's work it out.

TABLE VI. EXAMPLE OF "POINTLESS"

Talker	Contents
X	It's difficult to make a new question, isn't it?
Y	Why don't we pick the best question among three of us and submit it?
X	I think that's great!
Y	OK, let's do so.

the 3 different levels, "100 to 150", "150 to 200" and "200-250", most of those groups that falls into the highest level, "200-250", also falls into "Detailed discussion on important points".

D. Leader Function on Chat

From the chat logs, learners who took a leader role in the chat were identified, and the relationship between the learners' rank for the pre-tests within their group and the qualities of their discussions was evaluated.

Fig. 26 shows a result of the groups of 2 people while Fig. 27 shows a result of the groups of 3 people. Based on the results, in the groups of 2 people, when those who played a leader role have less academic ability than those who did not, their discussion tends to be well. In the group of 3 people, on the other hand, when those who had the best grade within their group played a leader role, their discussion tends to be well.

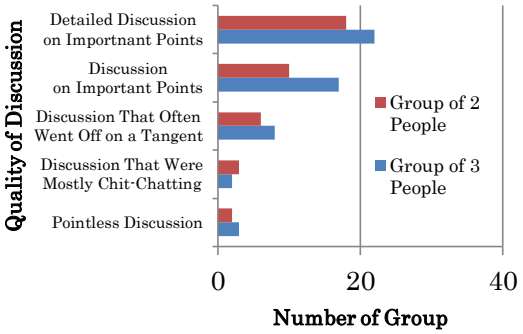


Figure 23. Quality of discussions and number of group

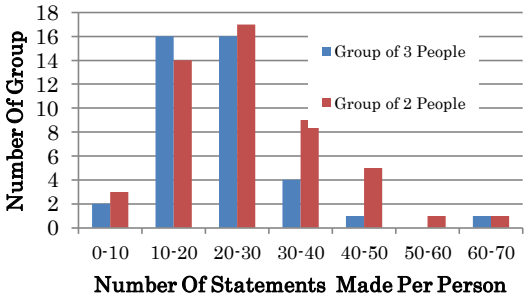


Figure 24. Number of statements made per person person within a group

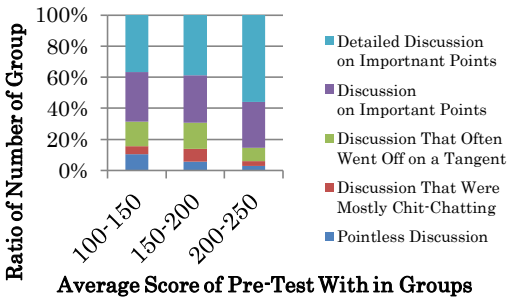


Figure 25. Pre-tests and quality of discussions

E. Evaluation of Group Assignments

In this experiment, since the assignments that are submitted individually and by groups are the same, these 3 patterns can be possible as re-submitted assignments: "Re-submitted after improving individual assignment", "Resubmitted the same individual assignments as is" and "Submitted completely new". Those assignments that were made completely new include the ones that combined several different assignments. Fig. 28 shows a distribution of the ways each group made their assignment. In both groups of 2 and 3 people, the results indicate most groups "Re-submitted after improving individual assignment".

"Re-submitted the same individual assignment as is" does not serve the meaning of collaborative learning, and it also means the collaborative script did not work well. Fig. 29 shows the quality of discussion being held by groups who

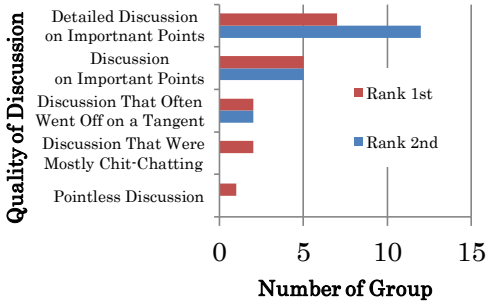


Figure 26. Leaders' rank in the group of 2 people

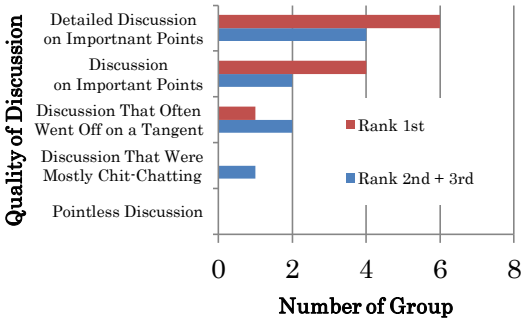


Figure 27. Leaders' rank in the groups of 3 people

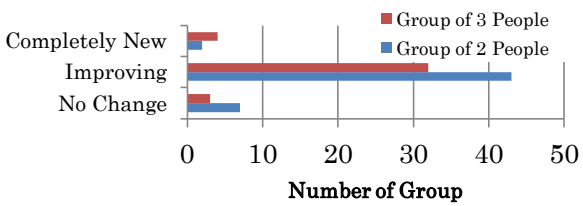


Figure 28. How they submitted group project

"Re-submitted the same individual assignment as is". Many of these groups had a discussion that was "Mostly chit-chatting" and "Pointless", so some type of scaffolding is necessary for them.

Table VII shows a standard for the group assignment, "Good", "Average" and "Bad", which are used for grading. Table VIII shows a comparison between the evaluation result and the qualities of the discussions. The evaluation was done by 1 teaching staff who carried out the experiment. There were 2 different evaluators for this evaluator and the one who evaluated the qualities of the discussions. The result shows that the better the discussion quality is, the higher the assignment evaluation is.

Also, Table IX shows a comparison between evaluation results and how discussions were carried on. "Made new" had a higher ratio of "Good" whereas "No changes" did not have any "Good". As Fig. 18 suggests, "No changes" tends to result in "More chit-chatting" or "Pointless". These points indicate that increasing a quality of discussion can lead to "Improvement" and "Make from scratch" with assignments highly scored.

VIII. SUMMARY AND FUTURE ISSUES

This section recapitulates the findings of this study and suggests briefly some future issues.

A. *Summary*

Supposing a situation where a face-to-face learning is impossible, we developed a CSCL system which can form many small groups for the online collaborative learning, and then the question-posing collaborative script based on the reciprocal teaching method was implemented in the system.

Then, in the environment with 300 people, the automated group formation and the collaborative script were proved executable and effective.

(1) The learners felt that the mutual work using the collaborative script was effective. In fact, discussions through the chat were activated while keeping their quality high.

(2) Many groups improved their submitted individual assignment through discussions online. Those groups that held high quality discussions scored high on their group assignment.

(3) It is suggested that the activation of discussions depends on an academic ability of the learners who play a leader role within their group. However, depending on a group structure, higher (academic ability) does not necessarily mean good.

First, according to (1) and (2), the results showed that the design of the collaborative learning in this study was mostly appropriate.

Also, according to (3), it is important to identify the most suitable learners to play a leader role and assign them in each group. However, the characteristics of learners who should play a leader role cannot be selected based on their academic ability, such as scores of pre-tests. To resolve such issue, in the future, it is important to develop a method to identify learners with an ability to take a leader role from a pre-survey and activity logs.

On the other hand, when the collaborative script is executed in a class, it is important to plan for exceptional cases, such as students' no attendance. Collaborative script does not allow a progress of tasks to be flexible, so the script often gets non-executable when the learning environment is off from an original plan. In this experiment, there are learners who attended on the 1st day and missed the 2nd day, or learners who missed the 1st day and attended on the 2nd day, so there were many groups that could not make progress their learning as planned. Also, there were some time limitations, such as a deadline for submitting assignments, so there were groups that had to

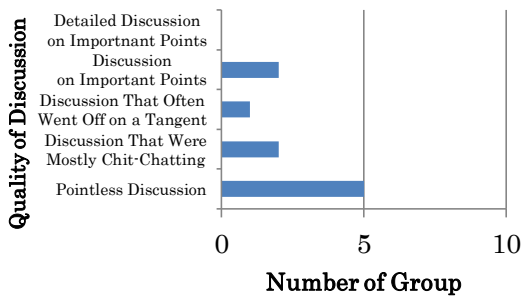


Figure 29. Quality of discussion held by groups without making changes

TABLE VII. EVALUATION STANDARD FOR PROJECT

Good	Complicated Question than the exercise shown in advance and an answer is right.
Average	Similar to the exercise shown in advance or equivalent in complexity, and a Answer is right
Bad	Similar to the exercise shown in advance or below equivalent in complexity, and an Answer is mistake

TABLE VIII. QUALITY OF DISCUSSION AND EVALUATION OF PROJECT BEING SUBMITTED

	Evaluation		
	Good	Avg	Bad
Detailed Discussion on Importnant Points	13	18	9
Discussion on Important Points	3	18	6
Often Went Off on a Tangent	2	5	7
Mostly Chit-Chatting		3	2
Pointless Discussion		2	4

TABLE IX. HOW DISCUSSIONS WERE MOVED FORWARD AND PROJECT EVALUATION RESULTS

	Evaluation		
	Good	Average	Bad
Completely New	2	3	1
Improving	16	38	22
No Change		5	5

submit without having sufficient discussions. Based on the above, executing a collaborative script needs some degree of flexibility depending on a learning environment and conditions of learners.

B. Future issues

In this study, the uniformed collaborative script was executed, but it is necessary to develop and practice collaborative script that is adaptable in groups in a way that the script changes flexibly depending on a group's characteristics and progress. In addition, future experiments have to examine what kind of difference manifests in the collaborative activities, depending on different communities or different learning agenda

Also, for the automated group formation, it is necessary to be capable of forming various groups based on learners' detailed characteristics being specified and to clarify characteristics of groups depending on learners included in the groups.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 23501117.

REFERENCES

[1] T. Inaba and K. Ando, "Development and assessment of CSCL system for large classrooms using collaborative script," in Proceedings of eLmL 2014, the Sixth International Conference on Mobile. Hybrid and On-line Learning, pp. 14-21, Spain, 23-27 March 2014.

[2] L. S. Vigotsky, Mind in society: The development of higher psychological processes, Cambridge: Havard University Press, 1978.

[3] J. Lave and E. Wenger, Situated learning: Legitimate peripheral participation, Cambridge: Cambridge University Press, 1991.

[4] H. Miyake and H. Shirouzu, Learning sciences and technology, Tokyo: The Society for the Promotion of the Open University of Japan, 2003.

[5] A. King, "Scipting collaborative learning processes: A collaborative perspective," in Scripting computer-supported collaborative learning, F. Fisher, I. Kollar, H. Mandl, and J. M. Haake, Eds. Springer, pp. 13-37, 2007.

[6] A. Weinberger, "Scripts for computer-supported collaborative learning: Effects of social and epistemic cooperation scripts on collaborative construction," Doctoral Dissertation, Ludwig-Maximilians-University, 2003.

[7] P. Bell, "Promoting students' argument construciton and collaborative debate in the classroom," in Internet environments for science education, M. C. Linn, E. A. Davis, and P.Bell, Eds. NJ: Erlbaum, pp. 114-144, 2004.

[8] R. C. Schank and R. P. Abelson, Scripts, plans, goals and understandings, Hillsdale, NJ: Erlbaum, 1977.

[9] D. F. Dansereau, "Cooperating learning strategies," in Learning and study strategies, E. T. Goetz and P. A. Alexander, Eds. Academic Press INC., pp. 103-120, 1989.

[10] A. O'Donell and D. F. Dansereau, "Scripted cooperation in student dyads: A method for analyzing and enhancing academic learning and performance," In Interaction in Cooperative groups: The theoretical anatomy of group learning, R. Herts-Lazarowitz, and N. Miller, Eds. New York: Cambidge University Press, pp. 120-141, 1992.

[11] I. Kollar, F. Fischer, and F. W. Hesse, "Collaboration scripts –A conceptual analysis," Educational Psychology Review, vol. 18, no. 2, pp. 159-185, 2006.

[12] L. Kobbe, A. Weinberger, P. Dillenbourg, A. Harrer, R. Hämäläinen and F. Fischer, "Specifying computer-supported collaboration scripts," International Journal of Computer-Supported Collaborative Learning, vol. 2, no. 2-3, pp. 211-224, 2007

[13] P. Dillenbourg and P. Jerman, "Designing interactive scripts," in Scripting computer-supported collaborative learning, F. Fisher, I. Kollar, H. Mandl, and J. M. Haake, Eds. Springer, pp. 275-301, 2007.

[14] P. Dillenbourg, "Over-scripting CSCL: The risks of blending collaborative learning with instructional design," In Three Worlds of CSCL, P. A. Kirschner, Eds. Heerlen: Open Univeristeit Nederland, pp. 61-91, 2002.

[15] N. Rummel and H. Spada, "Learning to collaborate: An instructional approach to promoting problem-solving in computer-mediated settings," Journal of the Learning Sciences vol. 14(2), 2005, pp. 201-241.

[16] A. Weinberger, Cscl scripts: Effects of social and epistemic scripts on computer-supported collaborative learning, Saarbrücken: VDM, 2008.

[17] F. Fischer, I. Kollar, K. Stegmann, and C. Wecker, "Toward a script theory of guidance in computer-supported collaborative learning," Educational Psychologist, vol. 48, no. 1, pp. 56-66, 2013.

[18] S. Takahashi, K. Ando, S. Matsunaga and T. Inaba: "Utilizing collaborative script that is adoptive to learners' characteristics to build and evaluate CSCL system," Proceedings of the 74th National Convention of IPSJ 4, pp. 815-816, 2012.

[19] A. S. Palinscar and A. L. Brown, "Reciprocal teaching of coprehension-fostering and comprehension-monitoring activities," Congnition and instruction vol.1, no. 2, pp. 117-175, 1984.

[20] M. Baker and K. Lund, "Promoting reflective interactions in a CSCL environment," Journal of Computer Assisted Learning, vol.13, pp. 175–193, 1997.

[21] H. R. Pfister and M. Mühlpfordt, "Supporting discourse in a synchronous learning environment: The learning protocol approach," in Proceedings of the conference on computer supported collaborative learning (CSCL) 2002 Conference, Laurence Erlbaum Associates, pp. 581–589, 2002.

A Novel Distributed Database Synchronization Approach with an Application to 3D Simulation

Martin Hoppen and Juergen Rossmann

Institute for Man-Machine Interaction
RWTH Aachen University
Ahornstrasse 55
52074 Aachen, Germany
Email: {hoppen,rossmann}@mmi.rwth-aachen.de

***Abstract*—3D (three-dimensional) simulation applications from various fields benefit from the usage of database technology. In contrast to the prevailing naive file-based approach, simulation models can be managed more efficiently, temporal databases can be used to log simulation runs, and active databases provide a means for communication. Thus, we use a central database to manage shared simulation models. To enable real-time access, each simulation client caches the model to its local runtime (in-memory) simulation database. For that purpose, a pairwise synchronization is needed between each runtime database and the central database. After a synchronization on schema level, each client replicates data on-demand. In this publication, we give a detailed description of our notification-based approach to keep master copies in sync with their replicate copies. The state of synchronization in between a pair of copies as well as allowed state transitions are comprehensively modeled using state machines. Moreover, we present three representative applications already using the approach, proving its practicability: City simulations, a Virtual Testbed for space robotics, and a forest inventory, management and simulation system.**

Keywords–Database Synchronization; 3D Simulation; Distributed Database; Applications.

I. INTRODUCTION

In this publication, we extend our previous work from [1]. In particular, we describe more aspects of our novel distributed database synchronization technique and give a more detailed insight into three application scenarios using the presented approach.

Simulation applications in general and 3D simulation applications in particular all follow the basic principle of applying simulation techniques to a corresponding model. Hence, the discipline is called modeling and simulation. A simulation model however needs some kind of data management. Up to now, files are still common for this task. In [2], we present a database-driven approach to overcome the associated disadvantages. Here, a central database is used to manage the shared simulation model, while simulation clients perform an on-demand replication of the model to their respective runtime database. The latter is an in-memory database providing the necessary real-time access. A revised version of this system was shown in [3], where the central database is even used as a communication hub to drive and log distributed 3D simulations.

In this paper, we add a detailed description of the notification-based synchronization approach used in this scenario. However, its specification should be preferably universal to allow for its adoption with different database systems. For that purpose, general requirements towards the two involved database systems – generically referred to as `ExtDB` (the central database) and `SimDB` (the runtime simulation database) – were compiled [4]. They incorporate methods adopted from Model-Driven Engineering (MDE) [5] and allow to use the concepts of the Unified Modeling Language (UML) [6] to give generalized method specifications for the different components of the overall approach [7]. Thus, in this publication, the synchronization approach will also be presented using UML metaclasses.

The synchronization approach relies on change notifications. Hence, ExtDB and SimDB need an according service. Using the notifications, the state of synchronization between both databases is monitored and modeled in a state machine for each pair of master and replicate copy. For resynchronization, transactions are scheduled and either executed or canceled out. Furthermore, notifications are used to confirm transactions and to detect change conflicts. A particular challenge in this scenario is to keep the state machine models ”stable”, i.e., not to miss or misinterpret notifications.

The approach is already used in different fields of applications, three of which are presented in detail in this paper: In various city and urban (distributed) 3D simulation scenarios, huge city models are stored in databases. Simulation clients use the presented approach to access the data and distribute changes like the movement of a car or a helicopter. In a Virtual Testbed for space robotics, planetary surveying, landing and exploration missions are developed and simulated using a shared world model. Different clients use the approach to access the model stored in a central database to deposit sensor data, extract maps, and utilize them for navigation and localization. Finally, in a large area forest inventory, management and simulation system, remote sensing data is used to extract semantic forest models managed in databases. Different stakeholders in the forestry sector can access these shared models to update, refine, simulate with, and analyze the data.

The rest of this paper is organized as follows: Section II presents work related to our own. In Section III, the foundations of the database-driven approach for 3D simulation are recapitulated. Section IV summarizes the system requirements

and the applied approach for method specifications using the UML metamodel. Both sections pave the way for the main Section V, where we present the notification-based synchronization approach. In Section VI, exemplary applications are shown. Finally, in Section VII, we conclude our work and present some future work.

II. Related Work

Regarding database synchronization for 3D simulation systems and similar software only few approaches can be found. In [8], a combination of scene-graph-based 3D clients with a federation of databases connected by the Common Object Request Broker Architecture (CORBA) is proposed. On client-side, a local object-oriented DBMS (OODBMS) provides an in-memory scene object cache connected to the federation using an Object Request Broker (ORB). Cached objects are bidirectionally replicated to the scene graph. Concurrency control among the federated databases and the local object caches allows multi user interaction between the clients.

A mobile Augmented Reality (AR) system combining distributed object management with object instantiation from databases is described in [9]. Objects are distributed shallowly by creating "ghost" copies retaining a master copy only at one site. Such a ghost is a non-fully replicated copy of its master allowing simplified object versions to be transmitted (e.g., with sufficient parameters for rendering). Changes to the master copy are pushed to all its ghosts. Remote systems can change a master copy by sending it a change request.

In [10], [11], a Virtual Reality (VR) system is combined with an OODBMS to provide VR as a multi-modal database interface. In [12], a revised version adds collaborative work support. For update propagation, VR clients issue changes to the shared virtual environment as transactions to the back-end they are connected to. After an interference check they are commited to the database and distributed by a separate notification service. The system uses transactions with regular ACID properties (e.g., for "Create box B") committed as a whole as well as special continuous transactions for object movements. For the latter, atomicity does not apply as movements are committed incrementally to frequently propagate updates.

The "Collaborative Urban Planner" described in [13] is based on the multi-user Virtual Environment system DeepMatrix [14], extended by a relational DBMS back-end providing persistency. Clients allow for so-called shared operations like "rotate object" that are send to the server for distribution and persistency. A server application provides concurrency control, message distribution and data management. It represents the single point of access to the database ensuring consistency among the clients' shared operations. The database primarily contains meta information on shared objects (position, texture).

In [15], a "Virtual Office Environment" contains 3D data and semantics managed by a DBMS to allow semantic-based queries and collaboration. Clients' actions are issued as queries to the shared database. Changes are distributed to all other clients, which adopt them locally.

A "shared mode" for database-driven collaboration is presented in [16]. In a chess application example with two players a shared database with the game's setting is alternately updated by the one client while being polled for changes by the other, which subsequently reflects the changes in his own virtual scene instance.

Compared to our approach, [8] comes close but lacks details and is only a proposal without known implementations. The ghosts in [9] may suffice for rendering but are to restricted for sophisticated simulation applications. Furthermore, not all objects are managed by the database. In [10], [11], [12], [15], only VR-specific data and operations are supported. [13] does not manage the model data itself using the database. Finally, the approach in [16] is similar to our own but only demonstrates a very limited type of change distribution. Altogether, no other approach offers a comparably tight integration of database technology into 3D software or simulation systems.

Similarities to our MDE-based approach for the general assessment of database compatibility can be found in generic model management. [17] introduces different generic schema operations like match, merge, translate, diff, and mapping composition. The work gives an overview but concentrates on tool support for semi-automatic mappings. Our own approach can be seen as an implementation of the "ModelGen" operator that automatically translates a schema from one metamodel into another, including mapping creation. However, in contrast, we provide an automatic mapping of schemata and a runtime approach instead of a static mapping.

Another implementation is provided in [18]. A pivotal supermodel is used to transform schema as well as data. In [19], the same system is extended to provide runtime transformations with read-only access. A similar approach is taken in [20] using a proprietary pivotal graph-based representation. [21] presents an approach for transforming schema and data between the Extensible Markup Language (XML) and the Structured Query Language (SQL). However, none of these approaches use standardized metamodeling and model transformation languages as used in our approach.

Besides these database-centric approaches, related work can also be found in the field of parallel and distributed simulation. Overviews can be found in [22] or [23]. In this field, approaches focus on the synchronization of events and time in simulation – somehow similar to our synchronization using change notifications. However, they cannot be directly applied to the presented problem of distributed data management. Here, events (change notifications) can only be monitored – they cannot be affected as in discrete event simulation.

III. Database-Driven 3D Simulation

Using a central database (ExtDB) to manage a shared simulation model has several advantages. In contrast to a classical file based approach, databases provide a very efficient data management, well-defined access points, e.g., using a query language or an Application Programming Interface (API), a consistent data schema for structured data, and concurrent access for multiple users. This allows to persist the current state of a 3D simulation model comprising its static (e.g., building, tree, work cell) as well as dynamic (e.g., vehicle, robot) parts. During a simulation run, the state of its model's dynamic parts changes. This is an inherent property of simulation. To capture this process over time, a temporal database [24] can be used. Here, any change to the simulation model causes the previous state's conservation as a version. Altogether, this also allows to persist the course of the simulation itself. Besides these more or less passive activities, a database can also be used as an active part of the simulation. One approach is to use it as an active communication hub. An active database [24] is needed

that can provide the necessary change notifications to inform clients of changes to the shared simulation model.

However, a steady, direct data exchange with ExtDB is not advisable for 3D simulation. This would lack real-time capabilities and impose a strong coupling on each and every component of the simulation system with the utilized database system. Instead, we use an approach that combines ExtDB with a local runtime database (SimDB) for each simulation client. The lower part of Figure 1 shows the principle structure of this approach for a single pair of ExtDB and SimDB instance. By replicating required contents from ExtDB to SimDB, the simulation system can use the cached copies and the nature of ExtDB can be hidden away.

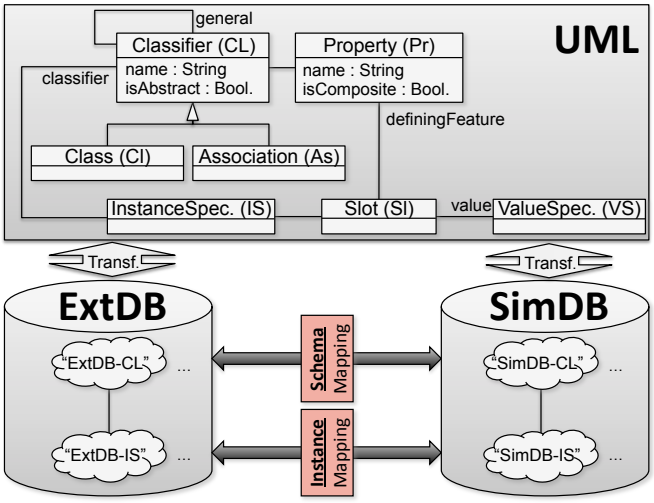


Figure 1. Principle structure of the approach for database-driven 3D simulation.

The two databases are synchronized on schema and data level. During the former, the schema description is transfered from ExtDB to SimDB so both systems ”speak the same language”. This builds up a schema mapping between the databases and is done once during system startup. Note however that this does not imply a semantic mapping like mapping an address represented by a single string to a fielded address representation (name, street, etc.). Instead, only the different modeling concepts (i.e., the utilized metaclasses) are mapped.

During runtime, data is loaded, i.e., replicated, from ExtDB to SimDB. Here, based on the schema mapping, the appropriate schema components are instantiated, values are copied, and an instance mapping (compare Figure 8) is stored to keep the relationship between master and replicate copy. Copies no longer required can also be unloaded, i.e., removed from SimDB provided they have not been changed. Changes are tracked and resynchronized to keep both master and replicate in sync. This is realized using notification services of ExtDB and SimDB. The approach is presented in detail in Section V. Besides for schema and instance data, synchronization can also be required on a semantic level. In functional data synchronization, the meaning of a modeled item is made available to the simulation system by translating it to a representation it can interpret. An example could be an engine modeled in the SEDRIS schema [25]. To allow a simulation of such a component it must be translated to the appropriate primitives of the simulation system.

IV. System Requirements

To generalize the approach system requirements were identified [4]. The aim is to make it universally available for different implementations of ExtDB and SimDB. First of all, a general compatibility of the two databases' modeling concepts is stipulated. For that purpose, both their metamodels are taken into account. A database's metamodel represents its abstract syntax, thus its modeling concepts. Metamodels shall not only comprise metaclasses for describing schema components like tables, classes, or attributes. They must also contain the corresponding instantiation concepts (e.g., metaclasses for rows, objects, or values). This is needed to also enable data synchronization. The two metamodels' compatibility can then be expressed with a model transformation, e.g., using the ATL Transformation Language (ATL) [26].

To provide a common basis for arbitrary database metamodels, a pivotal metamodel with transformations from and to both databases' metamodels is stipulated as well. The pivot's metaclasses can be used to indirectly refer to SimDB's or ExtDB's metaclasses using the demanded mapping. In the context of 3D simulation, Geographic Information Systems (GIS), Computer-Aided Design (CAD), or other 3D software, an object-oriented modeling is advisable, as such data usually consists of a huge number of hierarchically structured parts with interdependencies [24]. Thus, the UML (language unit classes) is a reasonable choice for a pivot. Figure 1 gives an overview. It also comprises the mainly utilized UML metaclasses. Altogether, this allows to generically refer to the structure of SimDB and ExtDB using UML concepts. Therefore, the method specification in the next section uses concepts like object, link, class, or property although including any database metamodel that can be mapped to the UML metamodel.

Note, however, that this mapping to UML structures is conceptually needed to show the databases' compatibility and to obtain a means for generalized method specifications. The actual implementation of the synchronization approach is done on API or query language level – in particular to ensure real-time capabilities.

The two databases are also required to provide a notification service. In terms of the UML metamodel, notifications shall provide information on object insertions and removals, on property updates, and on link insertions as well as removals. Furthermore, they must provide a reflection interface to access schema components and instantiate corresponding data. Finally, objects must be uniquely identifiable.

V. Notification-based Database Synchronization

This section represents the main contribution of this article: A detailed description of the notification-based synchronization approach.

A. Comparison with Distributed Databases

Following the definition in [24], the presented scenario, i.e., the combination of SimDB and ExtDB, would be a distributed database (DDB). Figure 2 depicts a classical DDB structure. Several databases build a virtual database that is transparently accessed via the DDBMS. In the example, a set of Door objects is horizontally fragmented, allocated to the different

databases and thus partially replicated. Similarly, our approach aims at transparency of the distribution. In Figure 3, it is depicted correspondingly. However, it is a special case, in which SimDB is a cache for ExtDB. Simulation clients access the shared simulation model only via SimDB. The nature and (for the most part) the existence of ExtDB are hidden away. The master copy of the simulation model is stored in ExtDB. An exception are local changes in a SimDB instance that are not yet synchronized to ExtDB, thus residing only at that client. In contrast, a classical DDB is accessed as a whole from the outside and the DDBMS hides away its distributive nature.

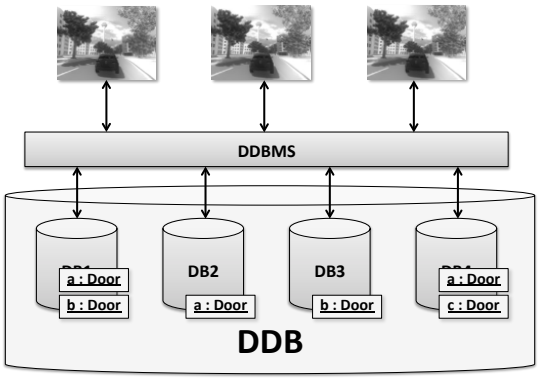


Figure 2. Classical distributed database with a centralized DDBMS.

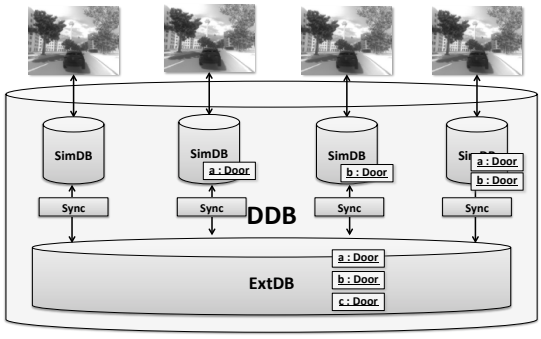


Figure 3. The presented approach can also be interpreted as a distributed database consisting of the central ExtDB and several SimDB instances.

Important DDB concepts are fragmentation, allocation and replication, as well as autonomy and heterogeneity. We use horizontal fragmentation splitting up object sets (but not objects themselves) between the central ExtDB and the connected SimDBs. All fragments are allocated to ExtDB. Further allocation, i.e., replication, to the different SimDBs is realized on-demand as shown in [7]. Thus, in the example in Figure 3, all Door objects are allocated to ExtDB and some of them are also allocated (i.e., replicated) to the connected SimDBs. While ExtDB is fully autonomous, SimDB is limited to the schema adopted from ExtDB. As both databases usually are different systems – e.g., SimDB is a runtime database – the assumed DDB is heterogeneous.

One or more instances of SimDB have a star-shaped connection to one instance of ExtDB. Changes are synchronized independently between each pair of SimDB and ExtDB. In the example in Figure 3, there are four SimDBs connected with their respective synchronization component to the central ExtDB. Differences in between such a pair are resynchronized periodically but not synchronously. Thus, we have a similar scenario as described in [27] for replication servers with asynchronous replication. However, in contrast to mobile databases, the connection is always kept alive and resynchronization is typically short-term. Furthermore, there is no global transaction or recovery manager. Changes to ExtDB by any client or to SimDB by any client component are committed without control of the synchronization component, which can merely monitor such changes. Thus, following durability (as in Atomicity, Consistency, Isolation, Durability (ACID)) they cannot be undone. Durability is important as an online (i.e., live) 3D simulation cannot be reset in the middle of a run.

B. Lock-free Approach for Simulation

One way to treat concurrent changes is an active concurrency control using locks. For distributed concurrency control, one approach is to choose a so called distinguished copy, which holds a representative lock for all its replicate copies [24]. In our case, the master copies in ExtDB could be adopted for this purpose as they are shared among all clients. However, locking is not recommendable here as acquiring locks would be time-consuming (as an ExtDB access would be necessary each time) and possible deadlocks may interrupt a running simulation.

Therefore, we developed a lock-free approach using notifications. For each pair of SimDB and ExtDB, the mechanism monitors changes by listening to the notifications. For resynchronization, it schedules transactions of the respective database. Due to the monitoring approach, they can only comprise a single data operation. The approach is similar to optimistic concurrency control (OCC) [27]. However, transactions cannot be rolled back when changes are conflicting. Instead, conflicts are only implicitly resolved: The last client changing a value is given precedence. Altogether, it is crucial that the synchronization component always knows about the state of synchronization for each copy. However, besides resynchronization and passive monitoring, the mechanism cannot and must not intervene, e.g., by rejecting changes as mentioned above.

C. Notifications and Transactions

For each pair of SimDB and ExtDB, a change tracking component connects to the notification services of SimDB for so-called *internal* notifications and of ExtDB for so-called *external* notifications. Notifications include insertions and removals of objects and links, as well as updates of object properties. A link between objects can only be removed or inserted but not updated, as its identity is only derived from the connected objects (and the corresponding association on schema level).

For the sake of simplicity, external notifications from ExtDB are abbreviated as extInsert, extUpdate, and extRemove, internal notifications from SimDB as simInsert, simUpdate, and simRemove, accordingly (Table I).

During runtime, these notifications are evaluated. Depending on the current state of the corresponding pair of master and replicate copy represented by an instance mapping entry, a transaction may be scheduled that can later be used to resynchronize the detected change from the one to the

TABLE I. TYPES AND ABBREVIATIONS OF NOTIFICATIONS FROM SIMDB AND EXTDB.

	internal (SimDB)	external (ExtDB)
property update	simUpdate	extUpdate
instance insertion	simInsert	extInsert
instance removal	simRemove	extRemove

other database. A scheduled transaction comprises one data operation with its kind (insert, remove, or update), the affected instance (object or link) or its id, and for updates the affected property. Table II gives an overview over the utilized transactions and their abbreviations. A transaction for transferring a change from SimDB to ExtDB is called an *out-bound* transaction and will be abbreviated with the prefix *sim2ext*.

For example, when detecting an object insertion within SimDB by a simInsert notification, a new sim2extInsert out-bound transaction may be scheduled. Its (future) execution will insert an equivalent object of the corresponding ExtDB-Classifier (using the schema mapping) into ExtDB. Here, the current property values are retrieved from the SimDB object's slots and are replicated for the new ExtDB object. Finally, the new object complements the corresponding instance mapping entry with its identifier. This can be seen as the complementing operation to the loading of objects. Links are treated accordingly but without the need for property value replication. An instance's removal (object or link) from SimDB, notified by a simRemove notification, may lead to a sim2extRemove transaction whose (future) execution will remove the associated ExtDB instance. A simUpdate notification signals the change of a SimDB object's property and may be scheduled as a sim2extUpdate transaction to transmit the value change from SimDB to ExtDB. Similar to sim2extInsert transactions, a sim2extUpdate transaction's execution retrieves the current value of its corresponding property from SimDB and replicates it to ExtDB.

TABLE II. TYPES AND ABBREVIATIONS OF TRANSACTIONS BETWEEN SIMDB AND EXTDB.

	out-bound: SimDB→ExtDB	in-bound: ExtDB→SimDB
property update	sim2extUpdate	ext2simUpdate
instance insertion	sim2extInsert	ext2simInsert
instance removal	sim2extRemove	ext2simRemove

Accordingly, external notifications may lead to the scheduling of *in-bound* transactions for resynchronizing global changes from ExtDB to SimDB. They are prefixed by *ext2sim*: ext2simInsert, ext2simRemove, and ext2simUpdate. Responses to external notifications are mostly identical to their internal counterparts. However, due to the nature of SimDB being a cache for ExtDB, a variation applies when treating external insertions. New objects or links within ExtDB may be handled by different strategies. They may be ignored or subsequently taken into account by a loading transaction (ext2simInsert). In this paper, the latter approach is chosen. Alternatively, one could consider to reevaluate previously executed queries to determine the "interest" in the new instance.

D. Change Propagation – The Basic Idea

Figures 4–7 show the basic idea of the notification-based distributed synchronization using an example with a central ExtDB and two SimDB clients. A door object is replicated to two simulation databases. One client changes the door's state indicated by a notification. The change is synchronized to the central database where another notification is issued. The latter causes another synchronization of the change to the second client.

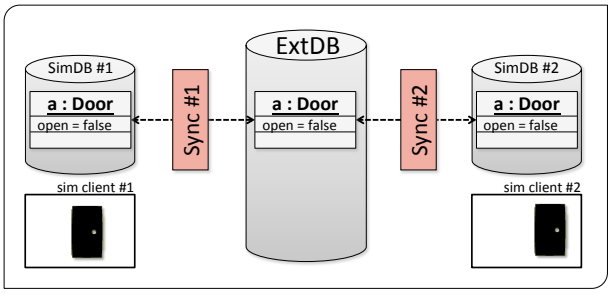


Figure 4. 1st step: Initial situation of a distributed synchronization example: all databases in sync.

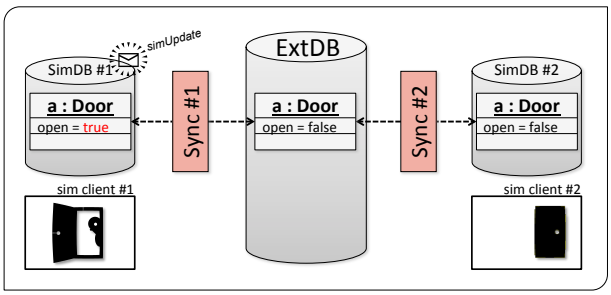


Figure 5. 2nd step: Client #1 changes the door's state; its SimDB issues a simUpdate notification.

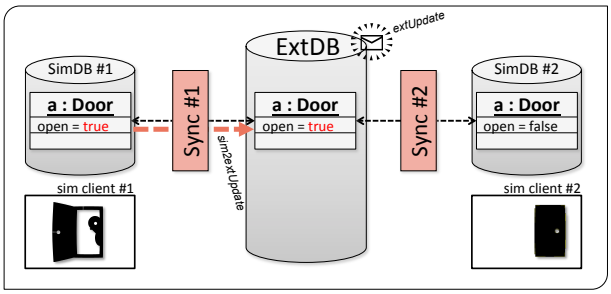


Figure 6. 3rd step: Sync component #1 responds using an out-bound sim2extUpdate transaction to synchronize the change to ExtDB, which in turn issues an extUpdate notification.

E. Modeling Synchronization With State Machines

Altogether, instance mapping entries (i.e., pairs of master and replicate copy) can be seen as to reside in a certain state of synchronization. Some examples are given in Figure 8: An object `a:Door` may exist in the central ExtDB without being loaded (i.e., replicated) to SimDB (i.e., no mapping exists), a replicated object `b:Door` may be unchanged (in sync), an object `c:Door` may only exist in SimDB (a transaction for its insertion in ExtDB is pending), a previously replicated object `d:Door` may be deleted in SimDB (a transaction

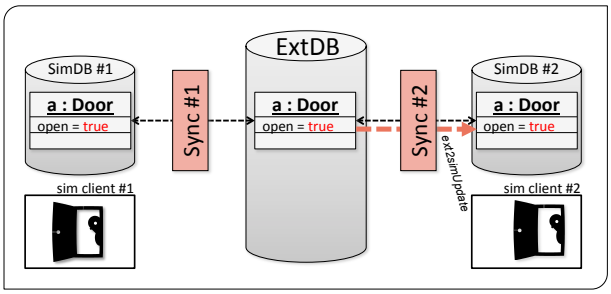


Figure 7. 4th step: Sync component #2 uses an in-bound ext2simUpdate transaction to adopt the change from ExtDB to SimDB #2.

for its deletion within ExtDB is pending), or a replicated object `e:Door`'s property may be changed within SimDB (a transaction for synchronization to ExtDB is pending).

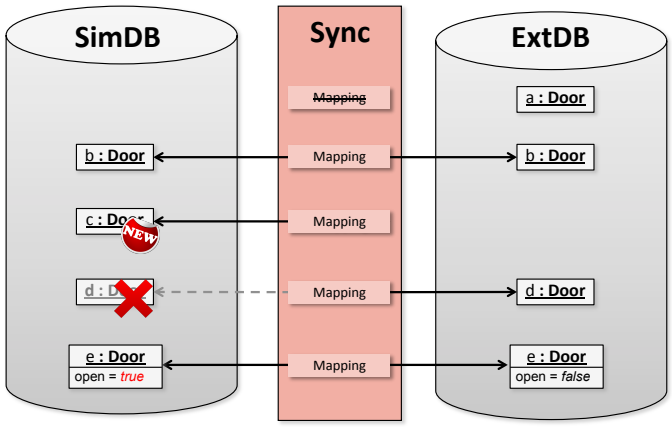


Figure 8. Exemplary mapping states between pairs of master and replicate copy.

This can be modeled as a state machine in statechart notation [28] for each object's or link's instance mapping entry. For objects, this state machine is given in Figure 11, for links in Figure 13. It may be in a synchronous state (*Synced*), a *Loading* or *Unloading* state, a state representing its absence or non-management (*NonManaged*), or a state of pending transaction (*ext2simInsertPending*, *ext2simRemovePending*, etc.). For update transactions, the synchronization states of an object's properties are concurrently modeled in the sub states of state *UpdatesPending* shown in Figure 12.

In these state machines, events comprise internal and external notifications, as well as some management events for object loading and unloading, failure thereof, and update completion. To simplify the state machine diagrams, any event undefined for a state shall trigger a transition to an omitted error state. Notification events are also implicitly filtered based on the related instance to match the considered instance mapping. Id est, the state machine for a certain instance mapping will only receive notifications for the corresponding instances from ExtDB or SimDB. Note that transitions depicted with italic text are special conditions dealt with in the Subsection V-G.

F. Event Handling Within the State Machines

An exemplary chain of events – depicted in Figure 9 – would be the insertion of a new door object into SimDB leading to a transition guarded by simInsert from the initial state NonManaged to state sim2extInsertPending shown in Figure 10.

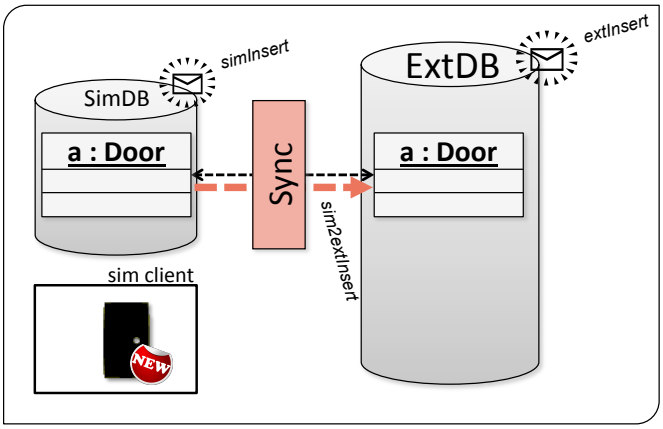


Figure 9. Exemplary insertion of a door object into SimDB and subsequent synchronization to ExtDB using a sim2extInsert transaction.

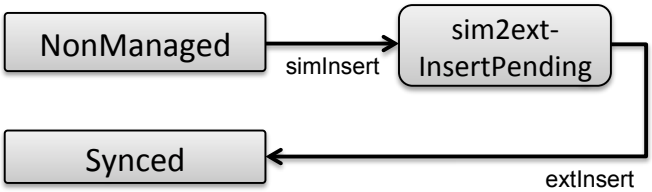


Figure 10. Excerpt from Figure 11 for the state transitions accompanying the exemplary insertion depicted in Figure 9.

In this example, a sim2extInsert transaction is scheduled for the new object. When the transaction is executed (see Subsection V-J), an equivalent object is inserted into ExtDB eventually causing the database to issue an extInsert notification. In turn, this event triggers a transition from the sim2extInsertPending to the Synced state. Thus, the extInsert event confirms the insertion into ExtDB and is used as a receipt to acknowledge a transaction's successful execution. This is especially useful for handling concurrent changes within SimDB and ExtDB occurring during other transaction's execution.

The receipt handling mechanism is also used to handle mutual changes that cancel each other out. An example are mutual removals: An instance is, e.g., first removed from ExtDB and subsequently from SimDB by independent processes. Thus, a previously scheduled ext2simRemove transaction with pending execution (in state ext2simRemovePending) is canceled out by the incoming simRemove notification for the same instance. The event causes a transition to the NonManaged state.

The same effect can be observed for the insertion of links. As before, links identify only by their member objects. In contrast to objects, they can be inserted identically but independently into both databases. In the state machine for links, this is represented by a transition from, e.g., sim2extInsertPending to Synced triggered by an extInsert for the identical link without having executed the scheduled transaction.

The state machines for objects and links differ only in some aspects. The latter allows an additional transition from the pending remove states back to the Synced state.

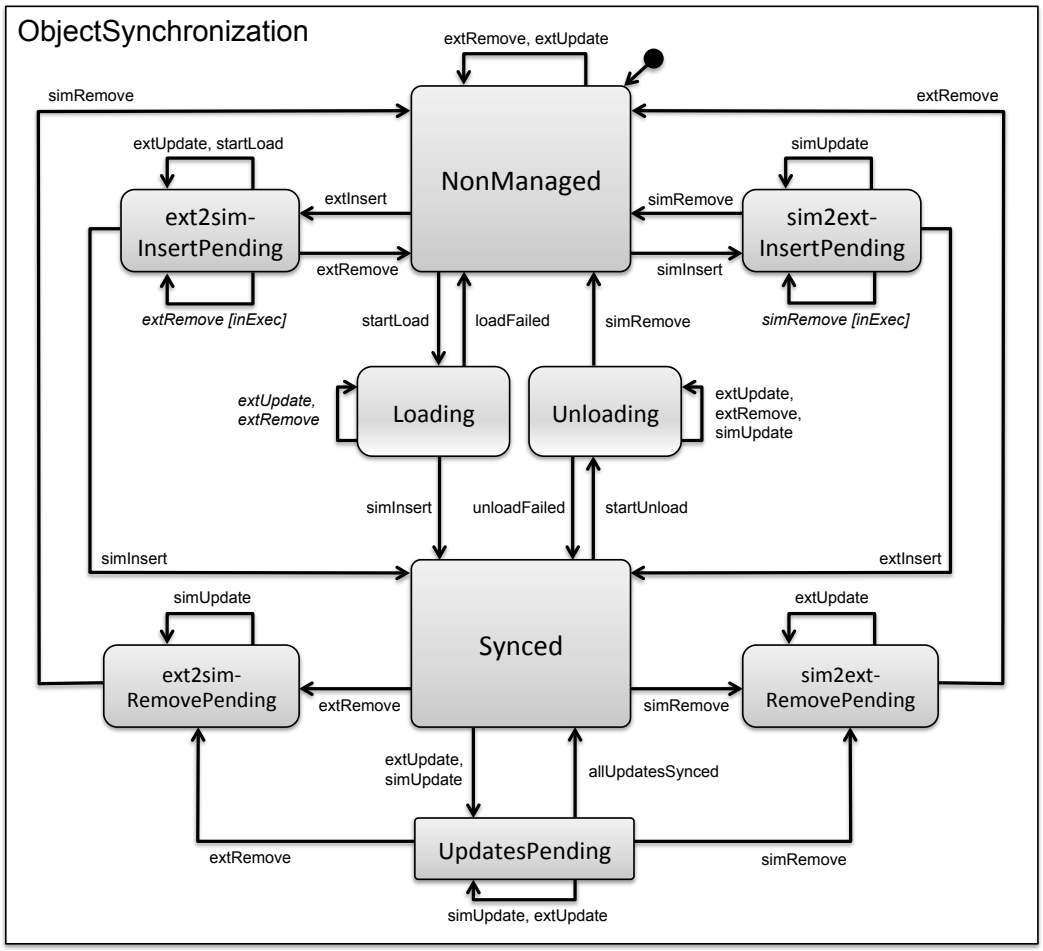


Figure 11. Synchronization states of an object's instance mapping.

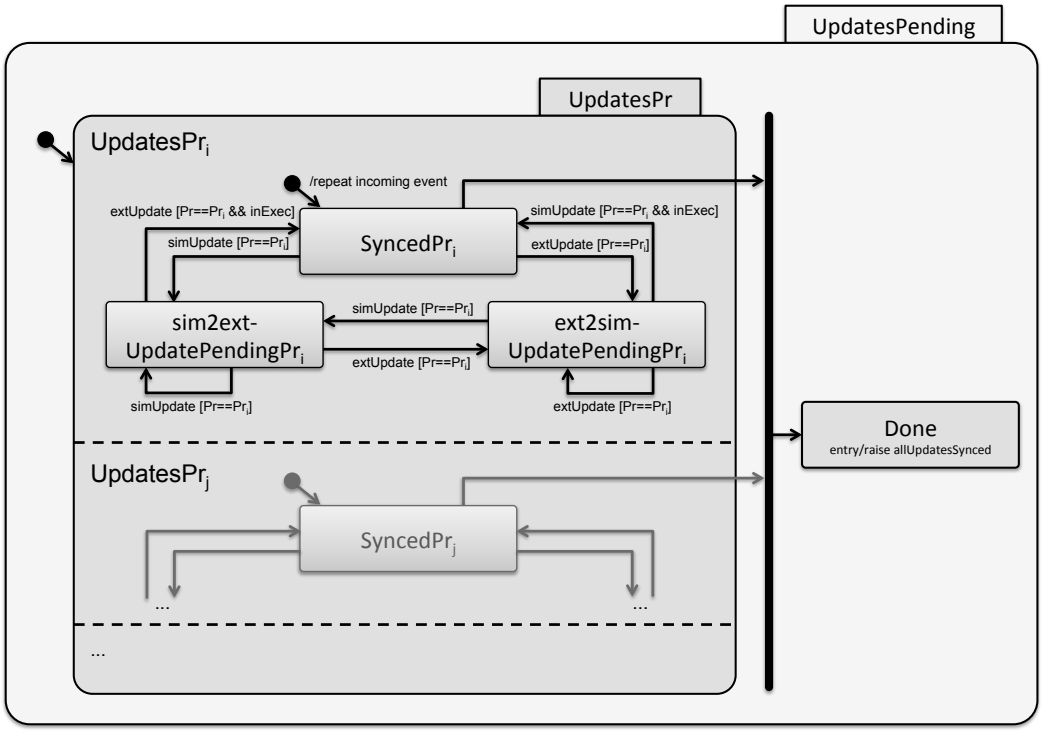


Figure 12. Sub structure of state UpdatesPending from Figure 11 for property updates.

In contrast to objects, an identical link can be reinserted after its removal. The former includes an additional update management for objects. The UpdatesPending state encapsulates a sub state structure for managing property updates (Figure 12). Primarily, it contains a super state UpdatesPr with concurrent regions for each of the object's properties, e.g., region UpdatesPr$_i$ for the object's i^{th} property. A region for Property Pr_i has three states representing an unchanged property value (SyncedPr$_i$), a property value changed within SimDB (sim2extUpdatePendingPr$_i$), and a property value changed within ExtDB (ext2simUpdatePendingPr$_i$). Internal transitions are triggered by update notifications (simUpdate and extUpdate) filtered for the corresponding property. To simplify modeling, the update event that caused the transition to UpdatesPending shall be repeated to initially activate the appropriate sub state, e.g., activate sim2extUpdatePendingPr$_i$ by repeating a simUpdate event. Further updates to the object's value for Pr_i can be ignored when they stem from the same database (i.e., both SimDB or both ExtDB). For example, in state ext2simUpdatePendingPr$_i$, further extUpdate notifications for Pr_i can be ignored as the new value has to be transferred to SimDB, anyway. However, a subsequent update to the same property from within SimDB causes a change conflict (see Subsection V-G). The modeled strategy is to give precedence to the more recently notified change. Thus, a transition to sim2extUpdatePendingPr$_i$ is triggered. When the transaction implicitly scheduled on entering one of the update states is executed, a notification is needed as a receipt. However, in contrast to insert or remove transactions, there is no "natural" counterpart for update transactions. An executed ext2simUpdate transaction causes a simUpdate notification that is indistinguishable from any other third party change. Thus, before execution, an "inExec" flag is set. For ext2simUpdate, the next simUpdate notification for Pr_i will trigger a transition back to the synced state of this property (the inExec flag will be reset). When all concurrent regions are in their respective synced state, a synchronized (in terms of concurrency) transition to the Done state is triggered (modeled by the vertical bar). On entering this state, the allUpdatesSynced event is raised triggering a transition from the super state UpdatesPending to the Synced state (see Figure 11).

In some situations, events may also be ignored. Within the state machines, this may be modeled as self-transitions. For example, in sim2extRemovePending, further extUpdate events from ExtDB can be ignored as the corresponding object will be removed from ExtDB, anyway. The same applies to pending insertions as property values will be replicated on execution time. For a pending ext2simInsert, the startLoad event has to be ignored, which will be emitted by the loading process used by the transaction. For non-managed instances within ExtDB, remove and update notifications can be ignored as they are of no interest to SimDB. Note that the sub regions in Figure 12 do not explicitly model ignored update events for properties $Pr \neq Pr_i$. Nevertheless, they shall be ignored and not cause a default transition to the omitted error state.

Besides change tracking, both state machines also contain states for the loading and unloading of objects or links. To announce a currently non-managed instance's loading, the methods presented in [7] shall raise an additional startLoading event. The instance remains in the Loading state either until its insertion into SimDB is acknowledged by an appropriate simUpdate receipt event or until the load process fails. Accordingly, before unloading a synchronized instance, a startUnload event shall be raised and the Unloading state is activated. Reaching this state, subsequent changes to the instance are ignored and consequently get lost. A simRemove notification serves as a receipt to acknowledge the unloading process and triggers a transition to the NonManaged state. Failing this, an unloadFailed event is raised to return to the Synced state.

G. Change Conflict Handling Using State Machines

As mentioned above, changes (insertions, removals, and updates) from ExtDB and SimDB may conflict when they occur to the same instance (and property) before executing the corresponding transaction. For example, in a city scenario, a building's street number is locally changed within SimDB causing a sim2extUpdate transaction. Before this change is made persistent and globally available within ExtDB by executing the transaction in a resynchronization run, the very same number is changed within ExtDB (e.g., by another simulation client). Following the strategy modeled above, the previous change is omitted and instead a new ext2simUpdate transaction is stored.

In general, different strategies to handle such situations could be thought of. First of all, conflicts can be avoided beforehand by giving only mutual exclusive write access to instances. This approach could be used in distributed simulation scenarios where separate objects are simulated by different clients without interaction. This can be managed by a superordinate simulation control. Avoiding the occurrence of conflicts could also be realized by explicitly locking changed instances or their property value in the respective other database. However, this may stall or even reset a simulation run as mentioned above.

Thus, a monitoring, i.e., reactive handling of change conflicts as mentioned above is inevitable. The presented methods' strategy is embedded in the given state machines. For change conflicts, two scenarios can be distinguished: A conflict may either occur *before* or *during* a transaction's execution. Before executing a transaction, conflict handling can be realized straightforward. It is modeled with simple transitions within the state machines. One example is the precedence for more recently notified updates as shown above. Another strategy is that object removals are final and thus "always win". Id est, a pending remove transaction for an object precedes all update events for the object. For pending object insertions, conflicts cannot occur as the corresponding object does not exist in the respective other database. In the context of links, the behavior slightly differs. Again, as a link identifies only by the objects it connects, one and the same link can be independently inserted and removed from both databases. Here, the same strategy as for the update of an object's property value applies: the more recently notified change precedes previous changes.

As long as a transaction is still pending, incoming events can always be processed by state transitions to reflect the relation between SimDB and ExtDB. In a resynchronization run, the current state of each state machine is evaluated (compare Subsection V-J). If a state with pending transaction $T1$ is determined $T1$ is executed. However, this decision is made independently at each client. A notification from a previously committed, conflicting transaction $T2$ may arrive just after $T1$'s execution is started. In some cases, $T1$ may

still be abortable. But the notification may just as well arrive when $T1$ commits. So, while native transactions of the utilized DBMSs themselves are usually isolated the decision to start a pending transaction is not. This limits transaction isolation (i.e., ACID properties) in the distributed system.

The same applies to the reading of property values. Objects can be removed, and links can be removed and inserted based only on the information from the corresponding notification. For object insertions and property updates however, the current state of the respective source database has to be retrieved as notifications themselves do not contain the corresponding values. Thus, when such a transaction is executed the source values may have already been changed by subsequent transactions whose notifications may either have not yet arrived or transaction execution may already have started as described above. This also limits transaction isolation.

Thus, a strategy had to be found for dealing with such situations. Otherwise, scenarios where a change in one database is neither reflected within the other database nor within the instance mapping's state machine may occur. For example, a property value is changed in ExtDB, but its instance mapping's state machine is in state Synced although SimDB still holds the previous value.

The primary instrument to handle such interfering changes is the aforementioned usage of notifications as receipts. For that purpose they must have the following features:

1) A notification's arrival guarantees the corresponding operation to be executed.
2) The order of arrival of a single database's notifications is identical to the execution order of the corresponding operations.
3) Between one running instance of SimDB and ExtDB there is at most one transaction being executed at a time (see Subsection V-J).

Based only on these assumptions, a conflict management can be stable. However, one should keep in mind:

1) A notification not yet received does not imply that the corresponding operation is not yet executed (notifications may be delayed).
2) On arrival of a notification, the *current* state within the database must not be consulted for further state transitions. By time of arrival it may already have been changed several times.
3) The order of arrival between notifications from ExtDB and notifications from SimDB is arbitrary.

Based on these considerations, a special event handling can be implemented to process the queued events after a transaction's execution. As stated above, the main problem are notifications arriving between the start of a transaction's execution and the arrival of the corresponding receipt notification. For a proper event handling, these events must sometimes be reordered. To be precise, they are captured and reinserted into the event queue just after the receipt event. This ensures their correct processing in terms of state transitions. The procedure is necessary for object or link insertions, link removals, object updates, and object or link loading. In the state machines, transitions with italic text particularly model this case. In the sub states of UpdatesPending, this highlighting is omitted as the same transitions are needed for standard and for this special event handling.

H. Exemplary Change Conflict Handling

One example for change conflict handling are updates (Figure 12). A property's update transaction can be examined separately as updates of different properties are independent from each other. Table III lists an exemplary sequence of events for some integer property and the associated actions, state machine states, values in SimDB and ExtDB, and emitted notifications. In the example, the local property's slot value in SimDB is updated several times even while changes are replicated to ExtDB. Notifications are used to ensure that all updates are reflected within the state machine's current state.

Initially (step #1), SimDB and ExtDB are in sync at value 10. The value in SimDB is changed to 20 (#2) and the corresponding simUpdate notification (a) triggers a state machine transition (#3). At some point in time, the client starts the resynchronization process (#4). Then, a first interfering update (#5) changes the value to 30. As property update notifications do not contain a value it must be retrieved from the respective database at transaction execution time (#6). Afterwards, a second interfering update (#7) changes the value to 40. In #8, the read value 30 is replicated to ExtDB. As mentioned above, the order, in which notifications from SimDB and ExtDB are received, is arbitrary. Thus, notifications simUpdate (a) and (b) may be processed first (#9, #10). As the "inExec" flag is set, all notifications are stored (instead of ignored without the "inExec" flag being set) until the corresponding receipt notification extUpdate is processed in #11. Subsequently, the flag is reset and both stored notifications are reinserted into the event queue. While the receipt notification eventually yields a transition back to the Synced state (#12), notification reinsertion causes the necessary transition back to the state of pending updates (#13) to replicate the value of 40 from SimDB to ExtDB. The additional simUpdate notification (c) only yields a self-transition (#14) as an update is already pending. Another resynchronization run would replicate the value to ExtDB starting at #15.

This approach to capture and reinsert notifications is needed as it is unknown whether an interfering update was done before (#5) or after (#7) reading the current value from SimDB in #6 to execute the sim2extUpdate transaction in #8. Note that when only interfering updates of the first type occur, the additional simUpdate notifications are in fact redundant. However, this is acceptable to guarantee that no updates are lost between SimDB and ExtDB. In case of interfering updates from other clients to ExtDB, additional extUpdate (instead of simUpdate) notifications are emitted. Here, notifications need not be stored as the first extUpdate notification is simply interpreted as the expected receipt and subsequent extUpdates yield normal state transitions. Finally, the same store-and-reinsert strategy is used similarly in the other use cases mentioned above (object insertions, link removals, and object or link loading).

Another example is a pending link insertion from SimDB to ExtDB (sim2extInsertPending) in Figure 13. After starting the transaction's execution, the link may concurrently be removed and reinserted into SimDB several times by different components of the simulation system. It may also be inserted into ExtDB from a third party client. In this case, sim2extInsert's execution will have no further effect. Subsequently, the link may even be removed and reinserted again within ExtDB. However, after the execution process, all corresponding (pos-

TABLE III. EXAMPLE OF A LOCAL INTERFERING UPDATE OF SOME INTEGER PROPERTY WITHIN A SINGLE SIMDB.

#	action	state machine	SimDB val.	ExtDB val.	notification
1	(initial state)	Synced	10	10	
2	update 10 → 20 in SimDB		20		simUpdate (a)
3	process event simUpdate (a)	→ UpdatesPending / sim2extUpdatePendingPr$_i$			
4	start resync	inExec := true			
5	update 20 → 30 in SimDB (1st interference)		30		simUpdate (b)
6	read current value from SimDB				
7	update 30 → 40 in SimDB (2nd interference)		40		simUpdate (c)
8	execute transaction sim2extUpdate			30	extUpdate
9	process event simUpdate (b)	[inExec=true] ⇒ store simUpdate (b)			
10	process event simUpdate (c)	[inExec=true] ⇒ store simUpdate (c)			
11	process event extUpdate	→ UpdatesPending / SyncedPr$_i$ → Done inExec := false reinsert simUpdate (b) in event queue reinsert simUpdate (c) in event queue			allUpdatesSynced
12	process event allUpdatesSynced	→ Synced			
13	process event simUpdate (b)	→ UpdatesPending / ext2simUpdatePendingPr$_i$			
14	process event simUpdate (c)	(self-transition)			
15	start resync ...	...			

sibly queued) events are processed until the receipt arrives. As the first extInsert notification is the receipt, no other extInsert and thus no extRemove notification can precede it. Hence, only simRemove and simInsert events have to be captured. They represent the current state of the link within SimDB. When the extInsert event arrives, these captured SimDB notifications are reinserted into the state machine's event queue just after the extInsert receipt. After the extInsert triggers a transition to the Synced state, the reinserted notification events are processed regularly. Note that it is irrelevant whether the link insertion in ExtDB actually originates from the successful execution of the sim2extInsert transaction or from a third party's operation. In both cases, the link is established and an extInsert notification is produced.

I. An Interim Conclusion

Regarding change tracking and conflicts, dependencies between different operations could also be a problem. Inserting an object usually also causes a parent link's insertion and removing an object cause a removal of all corresponding links and all (hierarchically) descendant objects. However, all these operations – dependent or not – cause the same kind of notifications and all interdependencies are resolved by the respective database itself. Thus, such dependent changes can be treated by the standard mechanisms and need no special handling.

Altogether, as mentioned above, this approach cannot avoid or fix conflicts but only detect them and react on them. However, the utilized SimDB and ExtDB themselves are not corrupted as they provide safe standard database access methods. Thus, only the distributed synchronization state must be kept free of corruptions. This is ensured by the presented approach.

J. Resynchronization

In resynchronization, all scheduled transactions are executed to bring the two databases back in sync. This process can be triggered in several ways. When the approach is applied in a collaborative scenario, it can be initiated manually. For immediate response from and to other users, it can also be automatically triggered after each transition to a state with pending transaction. In distributed simulation, typical access patterns include constantly repeated changes of the same few property values, e.g., a moving car and a moving helicopter. In such scenarios, transactions can be aggregated within short but arbitrary periods to lower the impact on traffic. However, this includes a trade-off between traffic and update rate.

Transactions are processed in groups of the same type to gain advantage from dependencies and optimize execution performance. An overview is given in the list below. At first, all ext2simRemove and sim2extRemove transactions for objects are executed to remove deprecated objects. As an optimization measure, this is done in reversed order of occurrence of the corresponding notifications. System requirements for the notification services specify the removal of object subtrees to be notified in bottom-up order. Additionally, subtrees may be removed manually in terms of further subtrees piece by piece. Thus, the transaction for a removal of the ExtDB root object of a completely removed subtree will always be preceded by all its descendant objects. Processing transactions in reversed order, the corresponding root object within SimDB will be deleted first and its instance mapping entry is removed. Due to the semantics of composite aggregation, this deletion will recursively delete all SimDB descendant objects in one operation (typically optimized by SimDB). Pending ext2simRemove transactions for descendant objects receive a notification for the removal of their target object, which serves as a receipt triggering a transition to the NonManaged state.

1) ext2simRemove and sim2extRemove for objects (in reverse order of occurrence)
2) ext2simInsert and sim2extInsert for objects
3) ext2simChange and sim2extChange (for objects)
4) ext2simRemove and sim2extRemove for links
5) ext2simInsert and sim2extInsert for links

Next, all ext2simInsert and sim2extInsert transactions for object insertions are processed. For each ext2simInsert the standard loading mechanism is used, including replicating

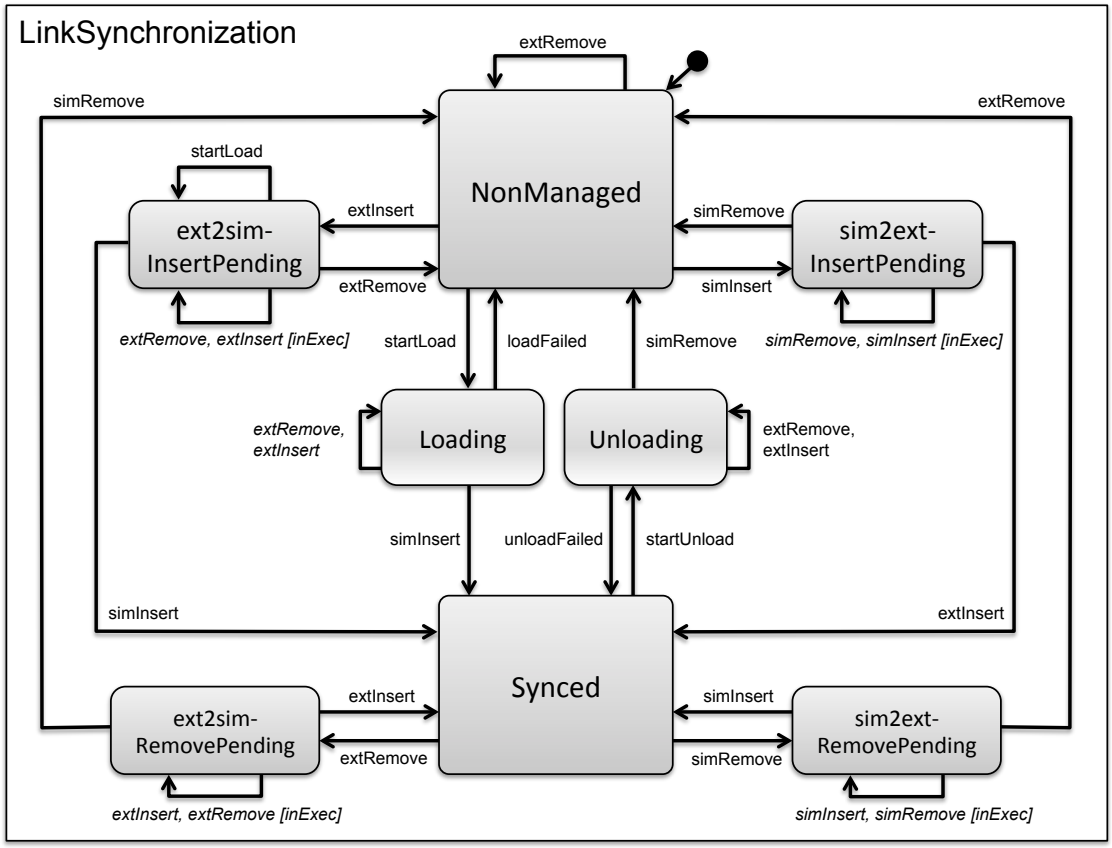


Figure 13. Synchronization states of a link's instance mapping.

current property values from the corresponding slots. Loading an object also includes loading all the object's ancestors to provide parent-child-relations within SimDB. Upon their loading, a startLoad event will be emitted to provoke a transition into the Loading state (for objects and parent links). Only for the object to be inserted itself and its parent link this event will be ignored. sim2extInsert transactions for objects make new objects globally available within ExtDB. The necessary operations can be seen as the counterpart to the loading operations.

Performing all object removals and insertions first assures further link operations are executed optimally. Beforehand, all ext2simUpdate and sim2extUpdate transactions are executed applying the current property values from ExtDB to SimDB and vice versa. In the next step, links are removed from SimDB and ExtDB by executing all corresponding ext2simRemove and sim2extRemove transactions. As with dependent object removals, each link already removed receives a receipt notification causing a transition to the NonManaged state. All remaining pending link removals are executed normally.

Finally, all ext2simInsert and sim2extInsert transactions are executed. The former are treated like their counterparts for objects by loading the corresponding link into SimDB using standard mechanisms. For parent links previously inserted in conjunction with an object insertion, receipt notifications may already have triggered a transition to the Synced state. As with link removals, all other link insertions are executed normally. When a designated link lacks a member object within the target database, the transaction will (detectably) fail to execute, reset the inExec flag, and stay in pending state. This may occur although object insertions are executed before link insertions. For example, an object and a link may be inserted while the resynchronization process has already executed all other object insertions. Likewise, a member object may be removed in the target database. However, the next resynchronization cycle will take care of this.

VI. Applications

Using the presented approach, different kinds of applications have already been realized.

A. City and Urban (Distributed) 3D Simulation

An application from the field of geo information systems are city simulations. Nowadays, a 3D city model exists for many cities in Germany like Dusseldorf or Stuttgart. Typically, these are often maintained by urban authorities, e.g., by land-registry. A widespread data model is the CityGML [29] standard, which is based on the Geography Markup Language (GML). A problem is to access, display, and use these huge data sets efficiently. Most tools can either access data in the original, highly semantic CityGML format but cannot efficiently display its 3D content or even do 3D simulation. Others can only efficiently use derived and optimized geometric representations, e.g., in VRML (Virtual Reality Modeling Language), which lack the semantic richness and need to be created by offline conversion. Using the presented approach

however, city data can be accessed in its original highly semantic schema making it available to a real-time capable 3D simulation system at the same time.

Prototypical 3D simulation scenarios include driving cars (with realistic physical correct behavior) or flying a helicopter through different city models (Figure 14). At the same time, the associated descriptive data can be accessed as the connection between geometry and object is never broken. As the proposed approach has read and write support (compare Section V), city data can even be updated from within the simulation system. In the context of CityGML, functional data synchronization has been used to supplement so-called implicit geometry representations with the corresponding referenced detail model. This was used for street furniture like street lights or benches or vegetation like trees. The implicit representation therefore contains references to external model files (e.g., in VRML format), which are subsequently loaded into the simulation system. For vegetation it may also just denote a tree type. Additionally, pose and size information are given, which are applied to the supplement.

Figure 14. Bringing large city models to life by combining database and simulation technology (data: Dusseldorf).

In another urban scenario, the presented methods have been used as the basis for a new approach to distributed 3D simulation using a central database. An important precondition is that the shared simulation model within ExtDB not only contains static model data like buildings, street furniture or vegetation. It also has to contain dynamic objects like a car or helicopter. A locally running simulation changes these dynamic objects (e.g., the cars position) within SimDB. Using the mechanisms described in Section V, these changes are synchronized to ExtDB, hence communicating the new state of the simulation model to the shared model. The database's notification service actively notifies any subscribed simulation client. Each client adopts the changes by synchronizing their own local replicate copies accordingly. All in all, this realizes a database-driven distributed 3D simulation.

In training scenarios using 3D simulation techniques – like driving or flying a virtual vessel or operating a virtual model of a machine – there is an interest in recording a simulation run. Recorded data can be used for replay, analysis, debriefing, and archival. This may be realized using external tools that log all interactions between the participants of the distributed simulation. Here, additional tools are needed to replay these logs. In the presented database-driven approach however, all changes are centrally routed through the shared model within ExtDB. Thus, as all intermediate states of the simulation are made persistent in the shared model, a simulation run can easily be captured by using a temporal database for ExtDB. The recorded time-stamped values not only represent a queryable 4D archive of the scenario. A simulation client can also be used in an off-line viewing mode to replay a simulation run step by step, allowing analysis and debriefing.

Using these two core concepts, a database-driven distributed 3D simulation application has been prototypically realized (Figures 15–16). The scenario consists of a heterogeneous shared model with a generic village in CityGML format as static model data and SEDRIS-based [25] car and helicopter models as dynamic model data. Data is loaded from a central database into two attached VEROSIM simulation clients. Here, simulation specific data is reconstructed using functional data synchronization. In particular, this includes relative transformations, which are synchronized bidirectionally to distribute movements of the dynamic objects. Dynamic parameters of the simulation model like physical characteristics of a car's drive were also encoded within the SEDRIS schema and reconstructed as supplementing structures in SimDB. However, these active components of the simulation are specially treated to configure responsibilities. I.e., the car shall only be actively simulated by one simulation client and only passively retraced within the other. The same applies to the helicopter vice versa. An id-based approach is used to configure these active and passive components per client. After replicating the data, the simulation is started in each client. Movements in one client are resynchronized to the central database and distributed to the respective other client. At the end of the simulation, the aforementioned usage of a temporal database allows an off-line replay of the simulation run.

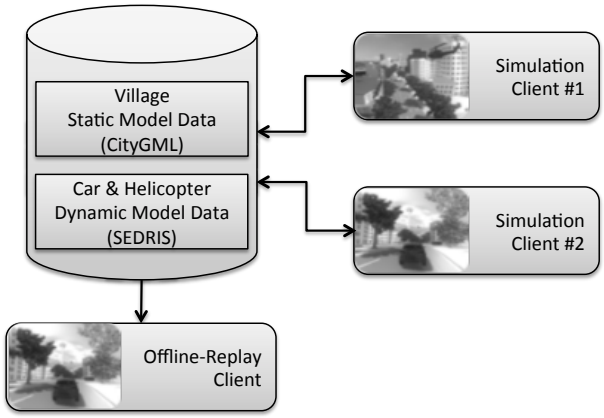


Figure 15. Architecture of the database-driven distributed 3D simulation application.

A performance evaluation yielded $0.7 - 1.3\text{ms}$ for single reading and $2.3 - 4.6\text{ms}$ for single writing operations (note that writing includes the versioning mechanism), which is acceptable for the given task. The central database uses a polling-based notification approach. It worked well but reached

Figure 16. The presented approach is used to actively drive a distributed simulation scenario.

its limits under heavy load making a push-based approach more advisable. For more details see [3].

Usually, such applications use amounts of files for data management combined with a decentralized communication infrastructure, e.g., based on the High Level Architecture (HLA) [30], and separate logging components are needed to archive a simulation. In contrast, we provide a more integrated approach. This avoids divergence between data management and the corresponding change distribution mechanism, no separate mechanism is needed to access logged data, and a consistent data schema provided by the central database is used throughout the distributed system.

B. Virtual Space Robotics Testbed

So-called Virtual Testbeds follow a holistic approach to 3D simulation. In contrast, conventional 3D simulation applications typically use very specialized techniques and focus on certain details. Virtual Testbeds however incorporate a diversity of problem aspects and their interactions into a single application, which integrates all relevant objects and parameters, their environment, and documentation. Similar to concepts for Product Data Management (PDM) systems [31], [32], they are also used in a much broader scope from design to testing, production and training. The Virtual Space Robotics Testbed (Figure 17) integrates the combined efforts of the research projects FastMap [33], [34], SELOK [35], [36], and Virtual Crater [37]. It provides means for the development, testing, and evaluation of robotic systems for space missions.

This Virtual Testbed's purpose is to support the development of future space missions. The three research projects pursue different aspects of this very same goal. The major goal of the research project FastMap is the automatic generation of navigation maps from images taken during the descent phase of a planetary landing mission. To calibrate the Virtual Testbed – in particular for camera and lighting – a physical planetary landing mockup is used. It consists of scaled realistic surface models and two robot arms carrying a light source and a camera. The flexibility of the Virtual Testbed even allows for its usage as a control system of this facility. The Virtual Testbed contains a model of a planetary descent scenario and additionally a virtual model of the physical mockup. All three scenarios (virtual mission, virtual mockup and physical mockup) are driven by the very same software components.

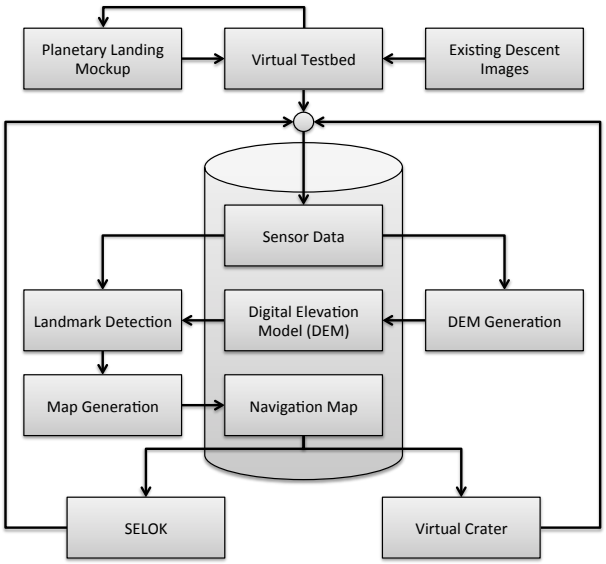


Figure 17. Structure of the projects FastMap, SELOK, and Virtual Crater in the Virtual Space Robotics Testbed.

A central database is used to manage a shared data model with a GML-based application schema. The central active database is used to make the shared model persistent and to communicate changes between the clients while deriving the navigation map. Hence, in this scenario, the approach is used for distributed data processing. The state of this distributed process is reflected by a central *mission* object holding a phase indicator. In each phase, a different client is responsible for finishing the phase (by incrementing it). Phase changes and all other data modifications including the insertion of new objects are actively communicated to all clients to drive the distributed approach.

In the first phase, a client (using the approach presented in this paper) processes sensor data, which can either stem from the physical mockup's camera or from one of the virtual scenarios (Figure 18). For each of these captured images, an object with the image's data is stored in the central database. When all images are captured, the phase is incremented. However, following a pipelining approach, as soon as the first image objects are stored in the database, the second client is notified and fetches these objects. A digital elevation model (DEM), i.e., a 3D model of the planet's ground, is constructed. Subsequently, DEM fragment objects holding the rasterized elevation data are also stored in the central database. When all captured images are processed, the phase is once again incremented. This phase change triggers the third client. Based on the images and the DEM, it applies different detection algorithms to create a map of landmarks like craters, rocks, or mountain tops. These landmarks are also stored in objects of the corresponding classes of the FastMap application schema. To end the distributed data processing, the detector client increments the phase of the mission for a last time.

Subsequently, the completed map is used as a basis for self-localization and navigation of mobile robots during the exploration of the planetary surface (Figure 19). This is the subject of the research project SELOK. Here, mobile robots use sensors like stereo cameras or laser scanners to capture

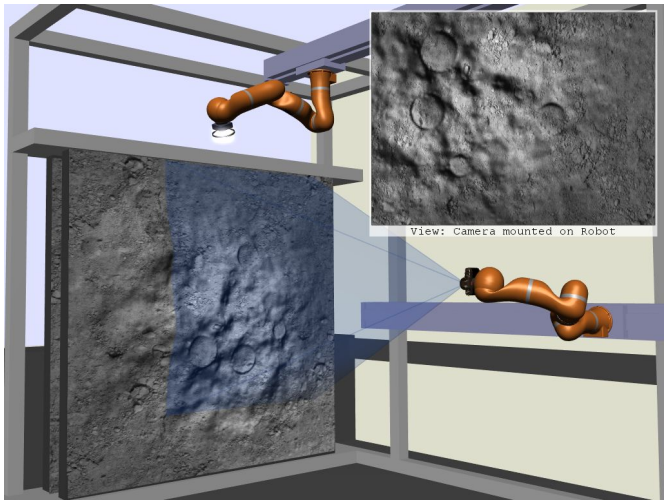


Figure 18. A simulation of a physical mockup for a planetary landing mission (project FastMap). The presented approach provides access to the shared world model.

their surroundings. The Virtual Testbed allows to flexibly combine and interchangeably use simulated and physical sensors. Within the data, landmarks like the aforementioned rocks or craters are detected. These so-called local landmarks are then matched against the global landmarks from the navigation map yielding a self-localization of the robot.

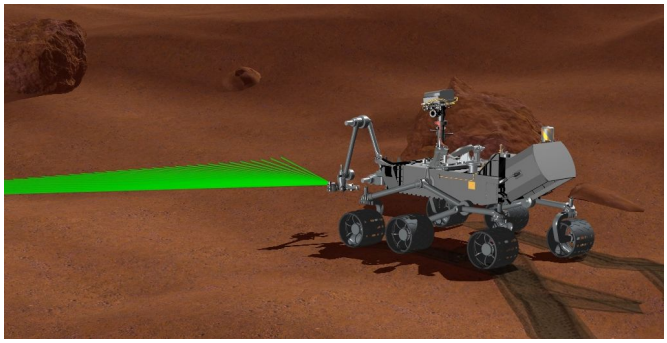


Figure 19. Planetary localization and navigation using the maps extracted after the landing phase. The presented approach is used to access the shared world model.

Finally, the third research project Virtual Crater is concerned with the development of a Virtual Testbed for mobile robots with planetary exploration missions. The testbed allows for a cost-efficient and realistic simulation of mobile robots in a virtual lunar environment. Here, they can be developed, programmed, tested, and optimized. Furthermore, the project comprises a physical testbed to compare and verify simulation results with reality. The results of Virtual Crater can be complemented by the navigation map and DEM from FastMap the self-localization methods from SELOK.

As this scenario provides similar conditions for the integration of the presented approach, evaluation results from Subsection VI-A are accordingly applicable.

C. Forest Inventory, Management and Simulation

In the field of forestry, the approach has successfully been employed for forest inventory, management and simulation. Most of the works where realized in the context of the research project Virtual Forest [38], [33], [39]. One of the core ideas of this project is a consistent, shared data model and data management in the Virtual Forest database (Figure 20). Provided to all stakeholders in this field, it facilitates the exploitation of know-how and synergies. Furthermore, it supports the transfer of industrial automation techniques to the forest industry.

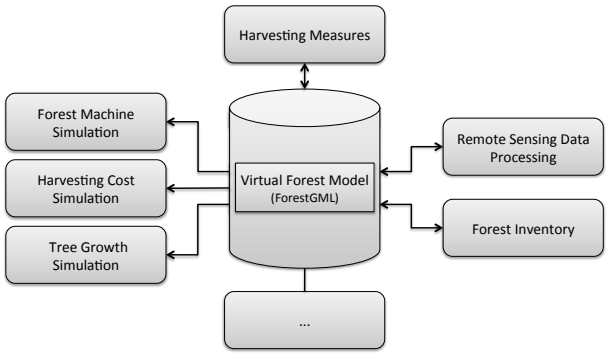


Figure 20. Architecture of the Virtual Forest scenario.

Integrating the database and all applications, the Virtual Forest constitutes a 4D geo-information system (GIS). It combines an object-oriented, semantic world modeling with 3D geo data with the fourth dimension of time in terms of a temporal database and simulation techniques. While the temporal database is used to preserve and access previous states, different simulation techniques are provided for prediction of future scenarios.

Forest inventory is the acquisition and management of environmental data in forestry. In the Virtual Forest, this is realized using a semi-automatic approach. Based on remote sensing and other data, algorithms for tree species classification, single tree delineation and attribution, and forest stand delineation and attribution are used to automatically process huge amounts of data. The results of these semantic world modeling processes are stored in the common Virtual Forest model. Expert users can then use them for quality inspection and data refinement. To improve their work, another important aspect are convenient user interfaces to algorithms and supportive tools (Figure 21).

Based on inventory data, other applications in the forestry context can be driven. There are simulation applications for decision support to predict harvesting costs or forest growth. Driver assistance or 3D simulations can be used to support and to train the usage of harvesters (see Figure 22) and forwarders in real world environments. Techniques for the automated feedback from harvesting allows for a permanent inventory. Here, results from felling measures are transferred back into the central model to keep it up-to-date. User-friendly query interfaces and reporting tools can help to evaluate and interpret the shared forest model. All in all, several applications for processing remote sensing data, semi-automatic inventory, simple forest information, forest planning, and wood production planning, and a Virtual Testbed have been developed. Furthermore, web technologies like portals and services provide the model to even more users.

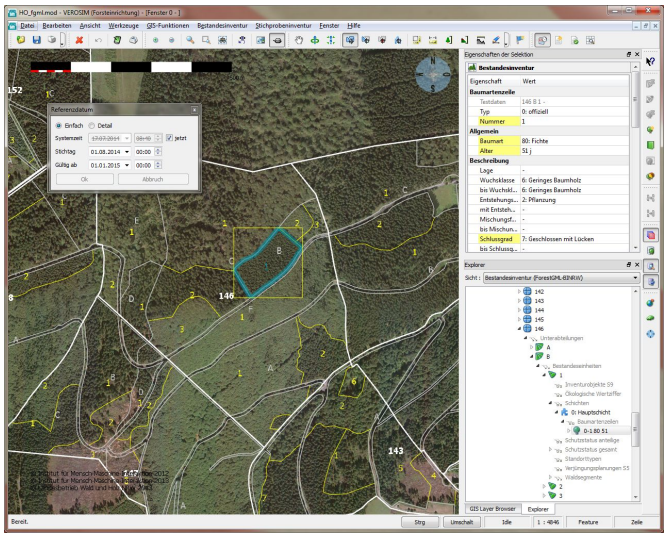


Figure 21. A forest inventory tool accessing a database with a detailed forest stand model.

In particular, the inventory tools have been used and tested in actual measures. In this context, they were evaluated for their real-world applicability – implicitly evaluating the presented approach utilized by these tools as well. Users attested the tools a very good performance and stability.

Figure 22. The presented approach is used for a harvester simulator to access the database managing the highly detailed forest model.

The 4D-GIS is realized using the simulation system VEROSIM and a geo database. The systems are combined by the presented prototype. Typically, inventory data is structured hierarchically to reflect, e.g., administrative units and contains spatial data. The presented approach allows multiple users to collaboratively work with a central database.

The data models employed for the Virtual Forest are all based on GML. A specialized base schema called ForestGML was developed to provide core data constructs to facilitate the Virtual Forest's usage in different areas, states, or administrations. Furthermore, GML as a base schema allows a standardized exchange of data using the corresponding standard web services like WFS (Web Feature Service) or WMS (Web Map Service). This guarantees interoperability between different clients connected to the Virtual Forest database.

VII. Conclusion and Future Work

We detailedly presented an approach for synchronizing a central database (ExtDB) with simulation databases (SimDB) as a basis for database-driven 3D simulation. After recapitulating our previously published background of the approach, the main contribution of this work is presented: A detailed description of the core method for distributed database synchronization. For each pair of master and replicate copy it manages the state of synchronization – modeled as a state machine. It is based on notifications provided by both databases. On the one hand, they are used to track the changes and schedule transactions for subsequent resynchronization. On the other hand, they are used as receipts to acknowledge transaction execution and to detect change conflicts. Compared to other methods for collaboration in 3D software systems, this approach provides a tight integration of advantages from the database field into simulation technology. To prove its practicability, examples of use from three different fields of application are presented in detail.

In future, we will examine further applications, e.g., from the field of industrial automation. Moreover, a porting of the approach to other database systems than the current prototypes will be reviewed. Finally, the integration of temporal databases will be examined in further detail, especially for valid time, bitemporal, or multi-temporal databases.

Acknowledgment

Virtual Forest: This project is co-financed by the European Union and the federal state of North Rhine-Westphalia, European Regional Development Fund (ERDF). Europe - Investing in our future.

References

[1] M. Hoppen and J. Rossmann, "A Database Synchronization Approach for 3D Simulation Systems," in DBKDA 2014,The 6th International Conference on Advances in Databases, Knowledge, and Data Applications, A. Schmidt, K. Nitta, and J. S. Iztok Savnik, Eds., Chamonix, France, 2014, pp. 84–91.

[2] J. Rossmann, M. Schluse, R. Waspe, and M. Hoppen, "Real-Time Capable Data Management Architecture for Database-Driven 3D Simulation Systems," in Database and Expert Systems Applications - 22nd International Conference, DEXA 2011, A. Hameurlain, S. W. Liddle, K.-D. Schewe, and X. Zhou, Eds. Toulouse, France: Springer, 2011, pp. 262–269.

[3] M. Hoppen, M. Schluse, J. Rossmann, and B. Weitzig, "Database-Driven Distributed 3D Simulation," in Proceedings of the 2012 Winter Simulation Conference, 2012, pp. 1–12.

[4] M. Hoppen, M. Schluse, and J. Rossmann, "A metamodel-based approach for generalizing requirements in database-driven 3D simulation (WIP)," in Proceedings of the Symposium on Theory of Modeling & Simulation - DEVS Integrative M&S Symposium, ser. DEVS 13. San Diego, CA, USA: Society for Computer Simulation International, 2013, pp. 3:1–3:6.

[5] M. Brambilla, J. Cabot, and M. Wimmer, Model-Driven Software Engineering in Practice, ser. Synthesis Lectures on Software Engineering. Morgan & Claypool Publishers, 2012.

[6] OMG, "Unified Modeling Language (UML)." [Online]. Available: http://www.uml.org 2014.12.01

[7] M. Hoppen, M. Schluse, and J. Rossmann, "Database-Driven 3D Simulation - A Method Specification Using The UML Metamodel," in 11th International Industrial Simulation Conference ISC 2013, V. Limère and E.-H. Aghezzaf, Eds., Ghent, Belgium, 2013, pp. 147–154.

[8] E. V. Schweber, "SQL3D - Escape from VRML Island," 1998. [Online]. Available: http://www.infomaniacs.com/SQL3D/SQL3D-Escape-From-VRML-Island.htm 2014.12.01

[9] S. Julier, Y. Baillot, M. Lanzagorta, D. Brown, and L. Rosenblum, "Bars: Battlefield augmented reality system," in NATO Symposium on Information Processing Techniques for Military Systems, 2000, pp. 9–11.

[10] Y. Masunaga and C. Watanabe, "Design and implementation of a multimodal user interface of the Virtual World Database system (VWDB)," in Proceedings Seventh International Conference on Database Systems for Advanced Applications. DASFAA 2001. IEEE Comput. Soc, 2001, pp. 294–301.

[11] Y. Masunaga, C. Watanabe, A. Osugi, and K. Satoh, "A New Database Technology for Cyberspace Applications," in Nontraditional Database Systems, Y. Kambayashi, M. Kitsuregawa, A. Makinouchi, S. Uemura, K. Tanaka, and Y. Masunaga, Eds. London: Taylor & Francis, 2002, ch. 1, pp. 1–14.

[12] C. Watanabe and Y. Masunaga, "VWDB2: A Network Virtual Reality System with a Database Function for a Shared Work Environment," in Information Systems and Databases, K. Tanaka, Ed., Tokyo, Japan, 2002, pp. 190–196.

[13] T. Manoharan, H. Taylor, and P. Gardiner, "A collaborative analysis tool for visualisation and interaction with spatial data," in Proceedings of the seventh international conference on 3D Web technology. ACM, 2002, pp. 75–83.

[14] G. Reitmayr, S. Carroll, A. Reitemeyer, and M. G. Wagner, "DeepMatrix - An open technology based virtual environment system," The Visual Computer, vol. 15, no. 7-8, Nov. 1999, pp. 395–412.

[15] K. Kaku, H. Minami, T. Tomii, and H. Nasu, "Proposal of Virtual Space Browser Enables Retrieval and Action with Semantics which is Shared by Multi Users," in 21st International Conference on Data Engineering Workshops (ICDEW'05). IEEE, Apr. 2005, pp. 1259–1259.

[16] K. Walczak and W. Cellary, "Building database applications of virtual reality with X-VRML," in Proceeding of the seventh international conference on 3D Web technology - Web3D '02. New York, New York, USA: ACM Press, Feb. 2002, pp. 111–120.

[17] P. A. Bernstein and S. Melnik, "Model management 2.0: manipulating richer mappings," in Proceedings of the 2007 ACM SIGMOD International Conference on Management of data - SIGMOD '07. New York, New York, USA: ACM Press, Jun. 2007, pp. 1–12.

[18] P. Atzeni, P. Cappellari, and P. Bernstein, "Model-Independent Schema and Data Translation," in Advances in Database Technology - EDBT 2006, ser. Lecture Notes in Computer Science, Y. Ioannidis, M. Scholl, J. Schmidt, F. Matthes, M. Hatzopoulos, K. Boehm, A. Kemper, T. Grust, and C. Boehm, Eds. Springer Berlin / Heidelberg, 2006, vol. 3896, pp. 368–385.

[19] P. Atzeni, L. Bellomarini, F. Bugiotti, and G. Gianforme, "A runtime approach to model-independent schema and data translation," in Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, ser. EDBT '09. New York, NY, USA: ACM, 2009, pp. 275–286.

[20] A. Smith and P. McBrien, "A Generic Data Level Implementation of ModelGen," in Sharing Data, Information and Knowledge, ser. Lecture Notes in Computer Science, A. Gray, K. Jeffery, and J. Shao, Eds. Springer Berlin / Heidelberg, 2008, vol. 5071, pp. 63–74.

[21] P. Berdaguer, A. Cunha, H. Pacheco, and J. Visser, "Coupled Schema Transformation and Data Conversion for XML and SQL," in Practical Aspects of Declarative Languages, ser. Lecture Notes in Computer Science, M. Hanus, Ed. Springer Berlin / Heidelberg, 2007, vol. 4354, pp. 290–304.

[22] R. M. Fujimoto, "Parallel and distributed simulation," in Proceedings of the 31st conference on Winter simulation Simulation—a bridge to the future - WSC '99, vol. 1. New York, New York, USA: ACM Press, Dec. 1999, pp. 122–131.

[23] K. S. Perumalla, "Parallel and distributed simulation: traditional techniques and recent advances," Dec. 2006, pp. 84–95.

[24] R. Elmasri and S. B. Navathe, Database Systems: Models, Languages, Design, And Application Programming, 6th ed. Prentice Hall International, 2010.

[25] SEDRIS, "SEDRIS." [Online]. Available: http://www.sedris.org 2014.12.01

[26] F. Jouault, F. Allilaire, J. Bézivin, and I. Kurtev, "ATL: A model transformation tool," Science of Computer Programming, vol. 72, no. 1-2, Jun. 2008, pp. 31–39.

[27] T. Connolly and C. Begg, Database systems: a practical approach to design, implementation, and management, 5th ed. Pearson Education (US), 2009.

[28] D. Harel, "Statecharts: a visual formalism for complex systems," Science of Computer Programming, vol. 8, no. 3, Jun. 1987, pp. 231–274.

[29] CityGML, "CityGML." [Online]. Available: http://www.citygml.org 2014.12.01

[30] Simulation Interoperability Standards Committee (SISC), "Standard for Modeling and Simulation High Level Architecture (HLA) IEEE 1516," 2000.

[31] Verein Deutscher Ingenieure (VDI), "VDI 2219 - Information technology in product development Introduction and economics of EDM/PDM Systems (Issue German/English)," Düsseldorf, 2002.

[32] U. Sendler, Das PLM-Kompendium: Referenzbuch des Produkt-Lebenszyklus-Managements (PLM compendium: reference book of product lifecycle management). Berlin: Springer, 2009.

[33] J. Rossmann, M. Schluse, R. Waspe, and R. Moshammer, "Simulation in the Woods: From Remote Sensing based Data Acquisition and Processing to Various Simulation Applications," in Proceedings of the 2011 Winter Simulation Conference, S. Jain, R. R. Creasey, J. Himmelspach, K. P. White, and M. Fu, Eds., 2011, pp. 984 – 996.

[34] J. Rossmann, T. Steil, and M. Springer, "Validating the Camera and Light Simulation of a Virtual Space Robotics Testbed by Means of Physical Mockup Data," in 11th International Symposium on Artificial Intelligence, Robotics and Automation in Space (i-SAIRAS), 2012, pp. 1–6.

[35] J. Rossmann, C. Schlette, M. Emde, and B. Sondermann, "Discussion of a Self-Localization and Navigation Unit for Mobile Robots in Extraterrestrial Environments," Artificial Intelligence, 2010, pp. 46–53.

[36] J. Rossmann, B. Sondermann, and M. Emde, "Virtual Testbeds for Planetary Exploration: The Self-Localization Aspect," in 11th Symposium on Advanced Space Technologies in Robotics and Automation (ASTRA), 2011, pp. 1–8.

[37] Y.-H. Yoo, T. Jung, M. Langosz, M. Rast, J. Rossmann, and F. Kirchner, "Developing a Virtual Environment for Extraterrestrial Legged Robots with Focus on Lunar Crater Exploration," in The 10th International Symposium on Artificial Intelligence,Robotics and Automation in Space (i-SAIRAS), 2010, pp. 206–213.

[38] J. Rossmann, M. Schluse, and A. Bücken, "The virtual forest - Space- and Robotics technology for the efficient and environmentally compatible growth-planing and mobilization of wood resources," FORMEC 08 - 41. International Symposium, 2008, pp. 3 – 12.

[39] J. Rossmann, M. Hoppen, and A. Bücken, "Semantic World Modelling and Data Management in a 4D Forest Simulation and Information System," ISPRS - International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences, vol. XL-2/W2, 2013, pp. 65–72.

In-Memory Distance Threshold Similarity Searches on Moving Object Trajectories

Michael Gowanlock
Department of Information and Computer Sciences and
NASA Astrobiology Institute
University of Hawai'i, Honolulu, HI, U.S.A.
Email: gowanloc@hawaii.edu

Henri Casanova
Department of Information and Computer Sciences
University of Hawai'i, Honolulu, HI, U.S.A.
Email: henric@hawaii.edu

***Abstract*—The need to query spatiotemporal databases that store trajectories of moving objects arises in a broad range of application domains. In this work, we focus on in-memory distance threshold searches which return all moving objects that are found within a given distance d of a fixed or moving object over a time interval. We propose algorithms to solve such searches efficiently, using an R-tree index to store trajectory data and two methods for filtering out trajectory segments so as to reduce segment processing time. We evaluate our algorithms on both real-world and synthetic in-memory trajectory datasets. Choosing an efficient trajectory splitting strategy to reduce index resolution increases the efficiency of distance threshold searches. Moreover, we demonstrate that distance threshold searches can be performed in parallel using a multithreaded implementation and we observe that high parallel efficiency (72.2%-85.7%) can be obtained. Interestingly, the traditional notion of considering good trajectory splits by minimizing the volume of hyperrectangular minimum bounding boxes (MBBs) so as to reduce index overlap is not well-suited to improve the performance of in-memory distance threshold searches.**

Keywords-query optimization; query parallelization; spatiotemporal databases; trajectory searches.

I. INTRODUCTION

Moving object databases (MODs) have gained attention as applications in several domains analyze trajectories of moving objects. Some examples include the movement patterns of animals in ecology studies, vehicles in traffic monitoring applications, stellar bodies in astrophysical simulations and the movement of objects in many other domains. Contributing to the motivation for MOD research is the proliferation of mobile devices that provide location information such as GPS tracking, which can provide short-term information to users, or whose data can be stored for subsequent processing. Regardless of how trajectory data is generated, trajectory similarity searches are used to gain insight into application domains, which attempt to find similarities between the properties of trajectories. Such similarities could be due to trajectories that cluster together over a time interval, or to trajectories with similar spatial properties, such as being short, or long, over a time interval. We focus on MODs that store historical trajectories [1], [2], [3], [4], [5], and that must support searches over subsets, or perhaps the full set, of the trajectory histories. In particular, we focus on two types of trajectory similarity searches, which we term *distance threshold searches*:

1) Find all trajectories within a distance d of a given static point over a time interval $[t_0, t_1]$.
2) Find all trajectories within a distance d of a given trajectory over a time interval $[t_0, t_1]$.

An example query of the first type would be to find all animals within a distance d of a water source within a day. An example query of the second type would be to find all police vehicles on patrol within a distance d of a moving stolen vehicle during an afternoon. We investigate efficient distance threshold searches on MODs, making the following contributions:

- We propose algorithms to solve the two types of in-memory distance threshold searches above.
- We make the case for using an in-memory R-tree index for storing trajectory line segments.
- Given a set of candidate line segments returned from the R-tree, we propose methods to filter out line segments that are not part of the query result set.
- We propose decreasing index resolution to exploit the trade-off between the volume occupied by the trajectories, the amount of index overlap, the number of entries in the index, and the number of candidate trajectory segments to process by exploring three trajectory splitting strategies.
- We demonstrate that, for in-memory searches, lower-bounding the index resolution is more important than minimizing the volume of MBBs, and thus index overlap.
- We parallelize the distance threshold search in a shared-memory environment using OpenMP and show that high parallel efficiency can be achieved.
- We evaluate our proposed algorithms using both real-world and synthetic datasets for both 3-D and 4-D trajectory data (i.e., the temporal dimension plus either 2 or 3 spatial dimensions).

This paper is organized as follows. Section II discusses related work. Section III defines the distance threshold search problem. Section IV discusses the indexing method. Section V details our algorithms. Section VI presents results from an initial performance evaluation of the two distance threshold searches outlined above. Section VII motivates, proposes, and evaluates methods for filtering the candidate line segments. Section VIII presents and evaluates methods for splitting trajectories to reduce index resolution for efficient query processing. Finally,

Section IX concludes the paper with a brief summary of findings and perspectives on future work.

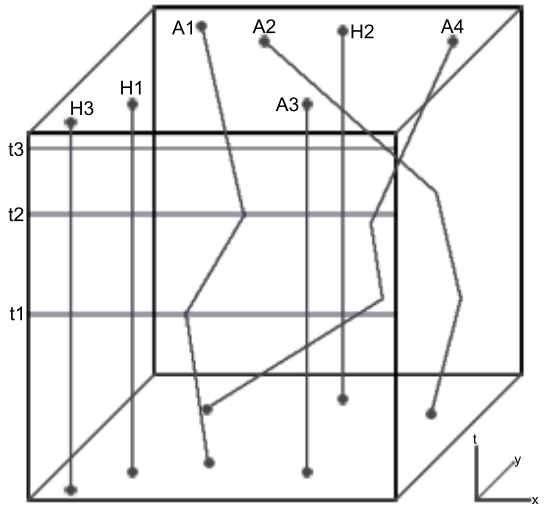


Figure 1. An illustration of trajectories and associated searches. Hospitals (H), ambulances (A), and time intervals between t_0 and t_4 are shown. Note that the vertical dimension refers to time in this example.

II. Related Work

A trajectory is a set of points traversed by an object over time in Euclidean space. In MODs, trajectories are stored as sets of spatiotemporal line segments. The majority of the literature on indexing spatiotemporal data utilizes R-tree data structures [6]. An R-tree indexes spatial and spatiotemporal data using MBBs. Each trajectory segment is contained in one MBB. Leaf nodes in the R-tree store pointers to MBBs and the segments they contain (coordinates, trajectory id). A non-leaf node stores the dimensions of the MBB that contains all the MBBs stored (at the leaf nodes) in the non-leaf node's sub-tree. Searches traverse the tree to find all (leaf) MBBs that overlap with a query MBB. Variations of the R-tree and systems have been proposed for efficient trajectory processing, such as TB-trees [7], STR-trees [7], 3DR-trees [8], SETI [9], SECONDO [10] and TrajStore [11].

In what follows we first review work on k Nearest Neighbors (kNN) searches. Although other types of searches have been studied (e.g., flocks [12], convoys [5], swarms [13]), kNN searches are the most related to distance threshold searches. We then review the (scarce) related work on distance threshold searches. Finally, we review work on the parallelization of searches in the MOD literature.

A. Nearest Neighbor Searches in Spatiotemporal Databases

Our work is related to, but as explained in later sections, also has major differences with the spatiotemporal kNN literature. We illustrate the typical kNN searches in Figure 1.

- $Q1$ Find the nearest hospital to hospital H_1 during the time interval [t_0,t_4], which results in hospital H_3.
- $Q2$ Find the nearest hospital to ambulance A_1 during the time interval [t_0,t_1], which results in hospital H_1.
- $Q3$ Find the nearest ambulance to ambulance A_2 during the time interval [t_1,t_4], which results in ambulance A_4.
- $Q4$ Find the nearest ambulance to ambulance A_4 *at any instant* in the time interval [t_0,t_4]; this results in multiple ambulances, since the query is continuous: ambulance A_3 in the interval [t_0,t_1), ambulance A_2 in the interval [t_1,t_3) and ambulance A_3 in the interval [t_3,t_4].

The example searches above are representative of the four main types of kNN searches that have been studied in the literature. The first type of search finds the nearest stationary data object to a static query object. An example is $Q1$ above. In [14], the authors propose a method that relies on the R-tree to perform NN searches, and then generalize their approach to handle kNN searches. The static objects are contained within a MBB. To process a search, MBBs are selected and then accessed from the R-tree to determine if a candidate NN is contained therein. To find the nearest neighbor two possible distance metrics are proposed: *MINDIST* and *MINMAXDIST*. These metrics are used for ordering and pruning the R-tree search.

The next type of search is moving query and static data, or the Continuous Nearest Neighbor (CNN) query [15], [16]. An example is $Q2$ above. In this search, the ambulance (or moving query) is continuously changing position relative to the hospitals; therefore, depending on the route, traffic, and other factors, the shortest distance to a hospital may change. The method in [15] employs a sampling technique on moving query objects, where query points are interpolated in between two sampled positions. The accuracy of this method is dependent on the sampling rate, which has the effect of making the method computationally expensive and can potentially return the wrong result. The CNN method developed by [16] avoids the computationally expensive drawbacks of [15]. Both works use an R-tree index.

The third type of search is moving query and moving data ($Q3$ above). There has been a considerable amount of work on this type of search (see for instance [17], [18], [19], [20], [21] for works published on this topic since 2005).

The last type of search is continuous moving query and continuous moving data (trajectories), which is the most related to this work. In $Q4$ above, multiple data points are returned as objects change over the time interval, in contrast to $Q3$ which is not continuous. These types of historical continuous searches have been investigated in [22], [23], [24], [10]. In comparison to the other NN variants, these searches propose new challenges: i) they are historical, meaning large segments can be processed; ii) they are continuous so that the candidate set changes over the time interval of the query. Opportunities arise for ordering and pruning the search efficiently, leading to several proposed new indexing techniques such as the TB-tree [7], the STR-tree [7], the 3DR-tree [8] and SETI [9].

The kNN search on historical continuous moving object trajectories is related to this work, but as detailed in Section IV, there are key differences that distinguish distance threshold searches from kNN searches. In particular, with distance threshold searches there is no possible pruning of the search space (while pruning is a main focus in the kNN literature).

B. Distance Threshold Searches

A distance threshold search can be seen as a kNN query with an unknown value of k. As a result, previous work on kNN searches (with known k) cannot be applied directly to distance threshold searches (with unknown k). To the best of our knowledge, other than our previous work [1], which we extend in this paper, the work in [4] is the only other work on distance threshold searches. In [4] the authors propose sequential, out-of-core solutions to the distance threshold search. They compare the performance of four implementations, one that is based on an R-tree index and three that utilize different plane-sweep approaches, and find that an adaptive plane-sweep approach performs the best in several experimental scenarios.

A difference between our work and that in [4], is that they only return whether trajectories are within a query distance d of query trajectory, and not the time interval within which this occurs. For example, over the time period [0,1], one trajectory may be within the query distance in the interval [0.1,0.2], and another trajectory within the interval between [0.4,0.9]. In contrast, our approach does provide these time intervals as this information can be useful in many application domains. One such example will be given in Section III-A. A more striking difference between [4] and this work is that we focus on in-memory databases, with a focus on efficient trajectory indexing strategies.

C. Parallelization of Searches in Spatial and Spatiotemporal Databases

The majority of the work in the literature on spatiotemporal databases has been in the context of out-of-core implementations where part of the database resides in memory and part of it on disk. Not much attention has been given to the parallelization of spatiotemporal similarity searches. To the best of our knowledge, the parallelization of distance threshold searches on moving object trajectories has not been investigated. In what follows we review relevant previous work on the parallelization of other searches.

The work in [25] provides a parallelization approach for finding patterns in a set of trajectories. The main contribution is an efficient way to decompose the computation and assign the trajectories to processors, so as to minimize computation and decrease communication costs. In [26], the authors propose a parallel solution for mining trajectories to find frequent movement patterns, or T-patterns [27]. They utilize the MapReduce framework in combination with a multi-resolution hierarchical grid to find patterns within the trajectory data. The importance of having multiple resolutions is that the level of resolution determines the types of patterns that may be found with the pattern identification algorithm. In [28], the authors propose two algorithms to search for the k most similar trajectories to a query trajectory where the trajectory database is distributed across a number of nodes. Their approach attempts to perform the similarity search, such that all of the relevant trajectory segments most similar to the query trajectory do not need to be sent across the network for evaluation, thereby reducing communication overhead. Finally, the work in [29] examines various indexing techniques for spatial and spatiotemporal data for use in the context of multi-core CPUs and many-core GPUs. Interestingly, the authors suggest that the traditional indexing techniques used for sequential executions may not be well-suited to emerging computer architectures.

III. Problem Statement

A. Motivating Example

One motivation application for this work is in the area of astrobiology/astronomy [30]. The field of astrobiology studies the origin, evolution, distribution, and future of life in the universe. The study of the habitability of the Earth suggests that life can exist in a multitude of environments. The past decade of exoplanet searches implies that the Milky Way, and hence the universe, hosts many rocky, low mass planets that may sustain complex life. The Galactic Habitable Zone is thought to be the region(s) of the Galaxy that may favor the development of complex life. With regards to long-term habitability, some of these regions may be inhospitable due to transient radiation events, such as supernovae explosions or close encounters with flyby stars that can gravitationally perturb planetary systems. Studying habitability thus entails solving the following two types of *distance threshold searches* on the trajectories of (possibly billions of) stars orbiting the Milky Way: (i) Find all stars within a distance d of a supernova explosion, i.e., a non-moving point over a time interval; and (ii) Find the stars, and corresponding time periods, that host a habitable planet and are within a distance d of all other stellar trajectories. These are exactly the distance threshold searches that we study in this work. Solving these searches for all stars with habitable planets will result in finding the duration of time flyby stars are within a distance threshold, for example 0.1 parsecs, of a habitable planetary system that is orbiting the Milky Way. As mentioned in Section II-B, it is useful to know the time intervals in which trajectories are within the query distance d. In this application, there is a difference between a flyby star that is only within the query distance for a short time interval, in comparison to a longer time interval, where the latter will mean a greater gravitational effect between the flyby star and planetary system. Additionally, the point distance search will find all stars hosting habitable planets sufficiently nearby a supernova such that the planets lose a significant fraction of the ozone in their atmospheres.

B. Problem Definition

Let D be a database of trajectories, where each trajectory T_i consists of n_i 4D (3 spatial + 1 temporal) line segments. Each line segment is defined by the following attributes: x_{start}, y_{start}, z_{start}, t_{start}, x_{end}, y_{end}, z_{end}, t_{end}, trajectory id, and segment id. These coordinates for each segment define the segment's MBB (note that the temporal dimension is treated in the same manner as the spatial dimensions). Linear interpolation is used to answer searches that lie between t_{start} and t_{end} of a given line segment.

We consider historical continuous *searches* for trajectories within a distance d of a *query* Q, where Q is a moving object's trajectory, Q_t, or a stationary point, Q_p. More specifically:

- DistTrajSearch_Q$_p$(D,Q_p,Q_{start},Q_{end}, d) searches D to find all trajectories that are within a distance d of a given query static point Q_p over the query time period [Q_{start},Q_{end}]. The query is continuous, such that the trajectories found may be within the distance threshold

d for a subinterval of the query time [Q_{start},Q_{end}]. For example, for a query Q_1 with a query time interval of [0,1], the search may return T_1 between [0.1,0.3] and T_2 between [0.2,0.6].

- DistTrajSearch_Q$_t$(D,Q_t,Q_{start},Q_{end}, d) is similar but searches for trajectories that are within a distance d of a query trajectory Q_t.

DistTrajSearch_Q$_p$ is a simpler case of DistTrajSearch_Q$_t$. We focus on developing an efficient approach for DistTrajSearch_Q$_t$, which can be reused as is for DistTrajSearch_Q$_p$. We present experimental results for both types of searches.

In all that follows, we consider *in-memory databases*, meaning that the database fits and is loaded in RAM once and for all. Distance threshold searches are relevant for scientific applications that are typically executed on high-performance computing platforms such as clusters. It is thus possible to partition the database and distribute it over a (possibly large) number of compute nodes so that the application does not require disk accesses. It is straightforward to parallelize distance threshold searches (replicate the query across all nodes, search the MOD independently at each node, and aggregate the obtained results). We leave the topic of distributed-memory searches for future work. Instead we focus on efficient in-memory processing at a single compute node, which is challenging and yet necessary for achieving efficient distributed-memory executions. Furthermore, as explained in Section IV, no criterion can be used to avoid index tree node accesses in distance threshold searches. Therefore, there are no possible I/O optimizations when (part of) the database resides on disk, which is another reason why we focus on the in-memory scenario.

IV. Trajectory Indexing

Given a distance threshold search for some query trajectory over some temporal extent, one considers all relevant query MBBs (those for the query trajectory segments). These query MBBs are *augmented* in all spatial dimensions by the threshold distance d. One then searches for the set of trajectory segment MBBs that overlap with the query MBBs, since these segments may be in the result set. Efficient indexing of the trajectory segment MBBs can thus lower query response time.

The most common approach is to store trajectory segments as MBBs in an index tree [22], [23], [24], [10]. Several index trees have been proposed (TB-tree [7], STR-tree [7], 3DR-tree [8]). Their main objective is to reduce the number of tree nodes visited during index traversals, using various pruning techniques (e.g., the *MINDIST* and *MINMAXDIST* metrics in [14]). While this is sensible for kNN searches, instead for distance threshold searches *there is no criterion for reducing the number of tree nodes that must be traversed*. This is because any MBB in the index that overlaps the query MBB may contain a line segment within the distance threshold, and thus must be returned as part of the candidate set.

Let us consider for instance the popular TB-tree, in which a leaf node stores only contiguous line segments that belong to the same trajectory and leaf nodes that store segments from the same trajectory are chained in a linked list. As a result, the TB-tree has high "temporal discrimination" (this terminology was introduced in [7]). Figure 2 shows a trajectory stored inside four leaf nodes within a TB-tree (each leaf node is shown as a bounding box). The curved and continuous appearance of the trajectory is because multiple line segments are stored together in each leaf node. By contrast, the R-tree simply stores in each leaf node trajectory segments that are spatially and temporally near each other, regardless of the individual trajectories. Figure 3 depicts an example with 4 segments belonging to 3 different trajectories that could be stored in a leaf node of an R-tree. For a distance threshold search, the number of TB-tree leaf nodes processed to perform the search could be arbitrarily high (since segment MBBs from many different trajectories can overlap the query MBB). Therefore, the TB-tree reduces the important R-tree property of overlap reduction; with an R-tree it may be sufficient to process only a few leaf nodes since each leaf node stores spatially close segments from multiple trajectories. For distance threshold searches, high spatial discrimination is likely to be more efficient than high temporal discrimination. Also, results in [7] show that the TB-tree performs better than the R-tree (for kNN searches) especially when the number of indexed entries is low; however, we are interested in large MODs (see Section III-A). We conclude that an R-tree index should be used for efficient distance threshold search processing.

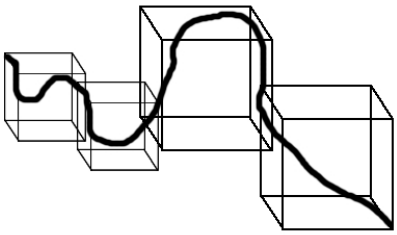


Figure 2. An example trajectory stored in different leaf nodes in a TB-tree.

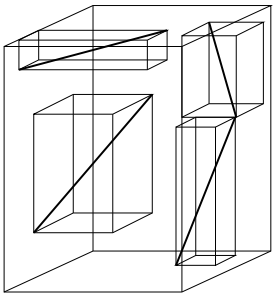


Figure 3. Four line segments belonging to three different trajectories within one leaf node of an R-tree.

V. Search Algorithm

We propose an algorithm, TrajDistSearch (Figure 4), to search for trajectories that are within a threshold distance of a query trajectory (defined as a set of connected trajectory segments over some temporal extent). All entry MBBs that overlap the query MBB are returned by the R-tree index and are then processed to determine the result set. More specifically, the algorithm takes as input an R-tree index, T, a query trajectory, Q, and a threshold distance, d. It returns a set of time intervals annotated by trajectory ids, corresponding to the interval of time during which a particular trajectory is within distance d of the query trajectory. After initializing the result set to the empty set (line 2), the algorithm loops over

```
 1: procedure TRAJDISTSEARCH (R-tree T, Query Q, double d)
 2:     resultSet ← ∅
 3:     for all querySegmentMBB in Q.MBBSet do
 4:         candidateSet ← T.Search(querySegmentMBB, d)
 5:         for all candidateMBB in candidateSet do
 6:             (entrySegment, querySegment) ← interpolate(
                    candidateMBB,querySegmentMBB)
 7:             timeInterval ← calcTimeInterval(
                    entrySegment,querySegment,d)
 8:             if timeInterval ≠ ∅ then
 9:                 resultSet ← resultsSet ∪ timeInterval
10:             end if
11:         end for
12:     end for
13:     return resultSet
14: end procedure
```

Figure 4. Pseudo-code for the TRAJDISTSEARCH algorithm (Section V).

all (augmented) MBBs that correspond to the segments of the query trajectory (line 3). For each such query MBB, the R-tree index is searched to obtain a set of candidate entry MBBs that overlap the query MBB (line 4). The algorithm then loops over all the candidates (line 5) and does the following. First, given the candidate entry MBB and the query MBB, it computes an entry trajectory segment and a query trajectory segment that span the same time interval (line 6). The algorithm then computes the interval of time during which these two trajectory segments are within a distance d of each other (line 7). This calculation involves computing the coefficients of and solving a degree two polynomial [10]. If this interval is non-empty, then it is annotated with the trajectory id and added to the result set (line 9). The overall result set is returned once all query MBBs have been processed (line 13). Note that for a static point search Q.MBBSet (line 3) would consist of a single (degenerate) MBB with a d extent in all spatial dimensions and some temporal extent, thus obviating the need for the outer loop. We call this simpler algorithm POINTDISTSEARCH.

VI. INITIAL EXPERIMENTAL EVALUATION

A. Datasets

Our first dataset, *Trucks* [31], is used in other MOD works [23], [24], [10]. It contains 276 trajectories corresponding to 50 trucks that travel in the Athens metropolitan area for 33 days. This is a 3-dimensional dataset (2 spatial + 1 temporal). Our second dataset is a class of 4-dimensional datasets (3 spatial + 1 temporal), *Galaxy*. These datasets contain the trajectories of stars moving in the Milky Way's gravitational field (see Section III-A). The largest *Galaxy* dataset consists of 1,000,000 trajectory segments corresponding to 2,500 trajectories of 400 timesteps each. Distances are expressed in kiloparsecs (kpc). Our third dataset is a class of 4-dimensional synthetic datasets, *Random*, with trajectories generated via random walks. An adjustable parameter, α, is used to control whether the trajectory is a straight line ($\alpha = 0$) or a Brownian motion trajectory ($\alpha = 1$). We vary α in 0.1 increments to produce 11 datasets for datasets containing between ∼1,000,000 and ∼5,000,000 segments. Trajectories with $\alpha = 0$ spans the largest spatial extent and trajectories with $\alpha = 1$ are the most localized. All trajectories have the same temporal extent but different start times. Other synthetic datasets exist, such as GSTD [32]. We do not use GSTD because it does not allow for 3-dimensional spatial trajectories.

TABLE I. CHARACTERISTICS OF DATASETS

Dataset	Trajec.	Entries
Trucks	276	112152
Galaxy-200k	500	200000
Galaxy-400k	1000	400000
Galaxy-600k	1500	600000
Galaxy-800k	2000	800000
Galaxy-1M	2500	1000000
Random-1M ($\alpha \in \{0, 0.1, \ldots, 1\}$)	2500	997500
Random-2M ($\alpha = 1$)	5000	1995000
Random-3M ($\alpha = 1$)	7500	2992500
Random-4M ($\alpha = 1$)	10000	3990000
Random-5M ($\alpha = 1$)	12500	4987500

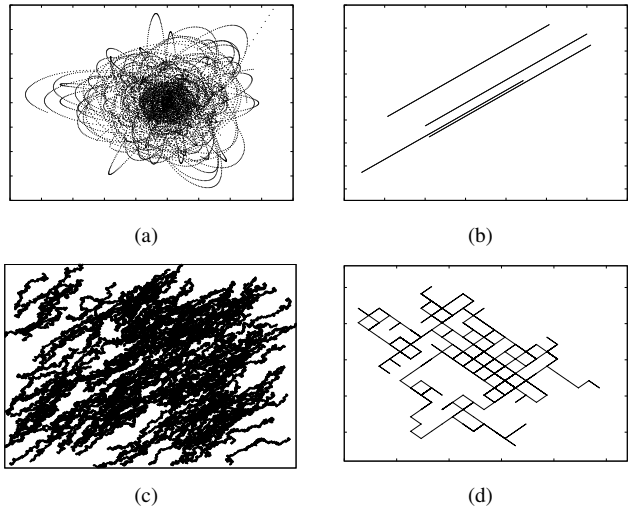


Figure 5. (a) *Galaxy* dataset: a sample of 30 trajectories, (b) 4 trajectories in the *Random* dataset with $\alpha = 0$, (c) 200 trajectories in the *Random* dataset with $\alpha = 0.8$, (d) a sample trajectory in the *Random* dataset with $\alpha = 1$.

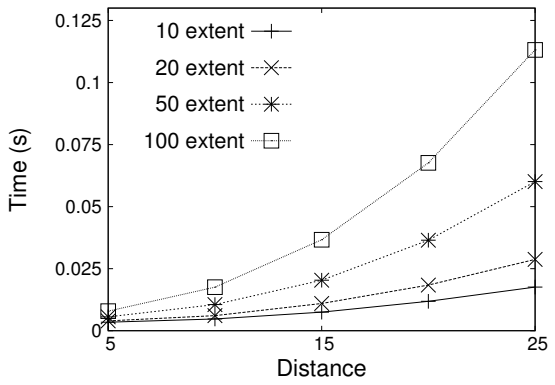
Figure 5 shows a 2-D illustration of the *Galaxy* and *Random* datasets. An illustration of *Trucks* can be found in previous works [23], [24]. Table I summarizes the main characteristics of each dataset. Our *Galaxy* and *Random* datasets are publicly available [33].

B. Experimental Methodology

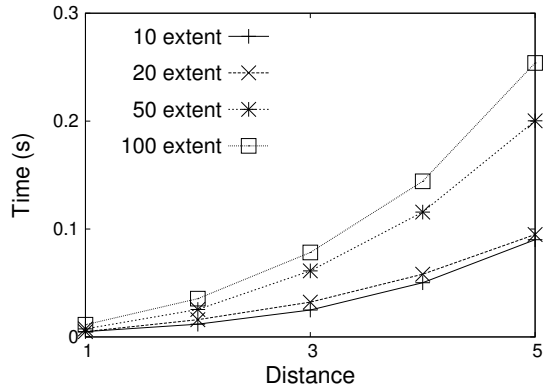
We have implemented algorithm TRAJDISTSEARCH in C++, reusing an existing R-tree implementation based on that initially developed by A. Guttman [6], and the code is publicly available [34]. We execute the sequential implementation on one core of a dedicated Intel Xeon X5660 processor, at 2.8 GHz, with 12 MB L3 cache and sufficient memory to store the entire index. In the multithreaded implementation, we show results up to 6 threads, which corresponds to the 6 cores on the CPU on the platform. We measure query response time averaged over 3 trials. The variation among the trials is negligible so that error bars in our results are not visible. We ignore the overhead of loading the R-tree from disk into memory, which can be done once before all query processing.

C. Static Point Search Performance

In this section, we assess the performance of POINTDISTSEARCH with the following searches:

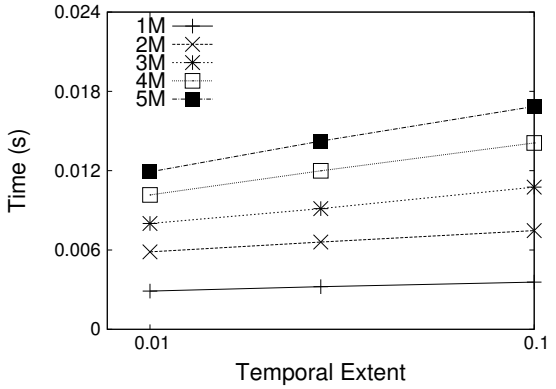


(a) *Random* $\alpha = 1$

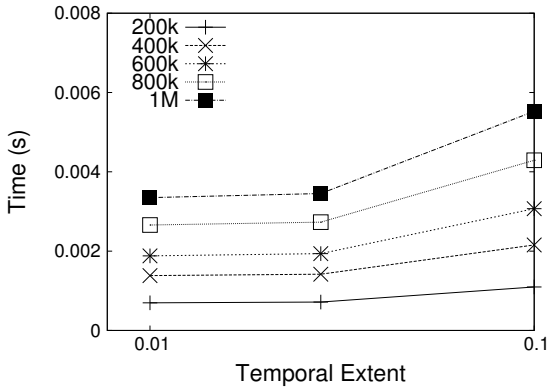


(b) *Galaxy*-1M

Figure 6. Query response time vs. threshold distance for 10%, 20%, 50% and 100% of the temporal extents of the trajectories in the datasets. (a) P1 using the *Random*-1M $\alpha = 1$ dataset; (b) the *Galaxy*-1M dataset with P2 (b).



(a) *Random* $\alpha = 1$



(b) *Galaxy* datasets

Figure 7. Query response time vs. various temporal extents of the trajectories in the datasets. (a) P3 using the *Random*-1M $\alpha = 1$ datasets; (b) P4 using the *Galaxy* datasets.

- P1: From the *Random*-1M $\alpha = 1$ dataset, 500 random points are selected with 10%, 20%, 50% and 100% of the temporal extent of the trajectories in the dataset, for various query distances.
- P2: Same as P1 but for the *Galaxy*-1M dataset.
- P3: From the *Random*-1M, 2M, 3M, 4M, 5M $\alpha = 1$ datasets, 500 random points are selected with 1%, 5%, and 10% of the temporal extent of the trajectories in the dataset, with query distance $d = 5$.
- P4: Same as S3 but for the *Galaxy*-200k, 400k, 600k, 800k, 1M datasets, where query distance $d = 1$.

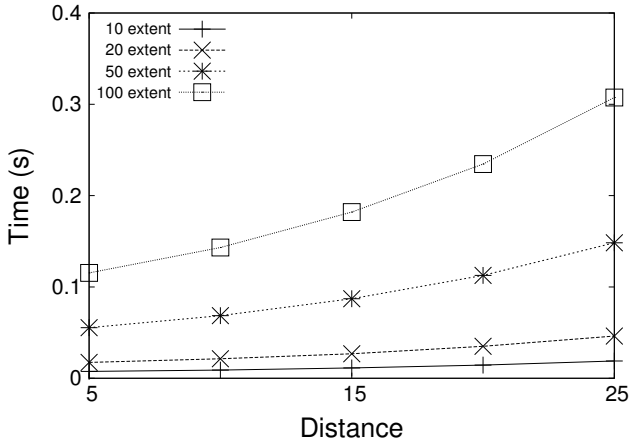
Figures 6 (a) and 6 (b) plot response time vs. query distance for P1 and P2 above. The response time increases superlinearly with the query distance and with the temporal extent. Figures 7 (a) and 7 (b) plot response time vs. temporal extent for P3 and P4 above, showing linear or superlinear growth in response time as the temporal extent increases. More specifically, Figure 7 (b) shows superlinear growth. This is because the trajectories in *Galaxy* are less constrained than in *Random*. We suspect that spatial under and overdensities of the trajectories in *Galaxy* may lead to searches that have qualitatively different behavior for different temporal extents. Next, we turn to our initial evaluation of trajectory searches.

D. Trajectory Search Performance

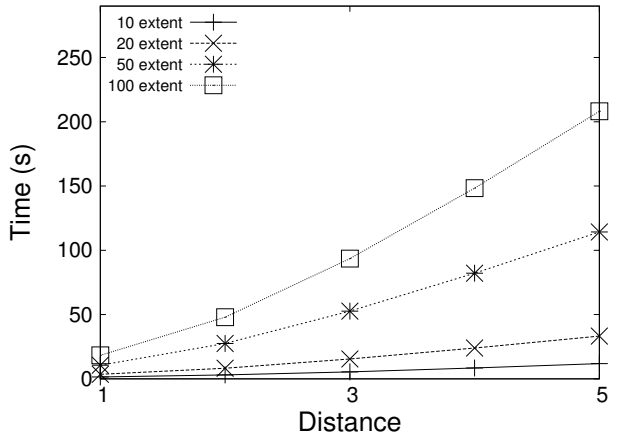
We measure the query response time of TRAJDISTSEARCH for the following sets of trajectory searches:

- S1: *Random*-1M dataset, $\alpha = 1$, 100 randomly selected query trajectories, processed for 10%, 20%, 50% and 100% of their temporal extents, with various query distances.
- S2: Same as S1 but for the *Galaxy*-1M dataset.
- S3: *Random*-1M, 2M, 3M, 4M and 5M datasets, $\alpha = 1$, 100 randomly selected query trajectories, processed for 100% of their temporal extent, with various query distances.
- S4: *Galaxy*-200k, 400k, 600k, 800k, 1M datasets, 100 randomly selected trajectories, processed for 1%, 5% and 10% of their temporal extents, with a fixed query distance $d = 1$.

Figures 8 (a) and 8 (b) plot response time vs. query distance for S1 and S2 above. The response time increases slightly superlinearly with the query distance and with the temporal extents. In other words, the R-tree search performance degrades gracefully as the search is more extensive. Figures 9 (a) and (b) show response time vs. query distance and temporal extent respectively, for S3 and S4 above. The response time increases slightly superlinearly as the query distance increases for S3, and roughly linearly as the temporal

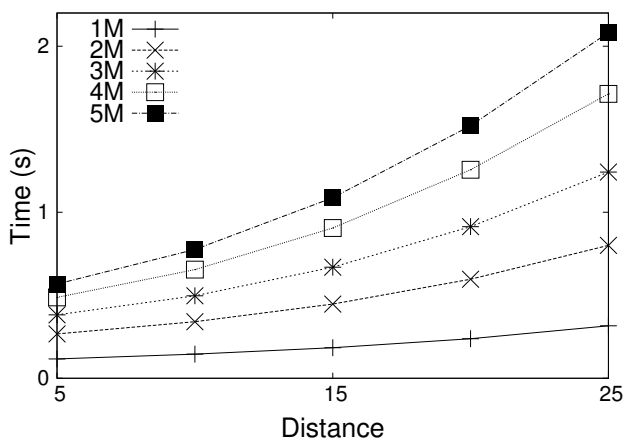


(a) *Random* $\alpha = 1$

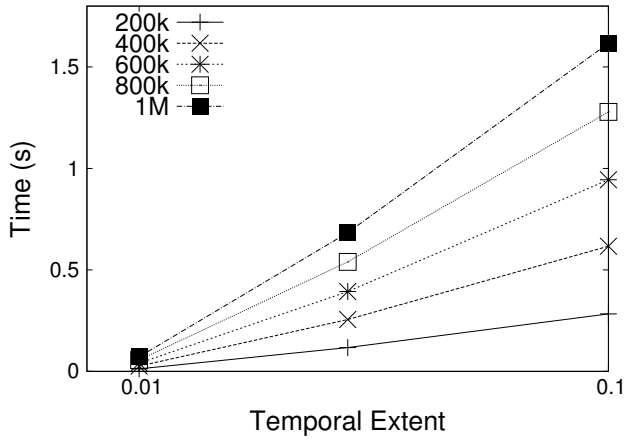


(b) *Galaxy*-1M

Figure 8. Query response time vs. threshold distance for 10%, 20%, 50% and 100% of the temporal extents of trajectories. (a) S1 using the *Random*-1M $\alpha = 1$ dataset; (b) S2 using the *Galaxy*-1M dataset.



(a) *Random* $\alpha = 1$



(b) *Galaxy* datasets

Figure 9. (a) Response time vs. threshold distances for various numbers of segments in the index using search S3. (b) Response time vs. temporal extent for various numbers of segments in the index using search S4.

extent increases for S4. Both of these figures show results for various dataset sizes. An important observation is that the response time degrades gracefully as the datasets increase in size. More interestingly, note that for a fixed temporal extent and a fixed query distance, a larger dataset means a higher trajectory density, and thus a higher degree of overlap in the R-tree index. In spite of this increasing overlap, the R-tree still delivers good performance. These trends are expected, as we see the performance of the algorithm degrade with increasing query distance, temporal extent, or dataset size. In the next sections we address optimizations to reduce response time further.

VII. Trajectory Segment Filtering

The results in the previous section show that PointDistSearch and TrajDistSearch maintain roughly consistent performance behavior over a range of search configurations (temporal extents, threshold distances, index sizes). In this and the next section, we explore approaches to reduce response time, using TrajDistSearch as our target.

At each iteration our algorithm computes the moving distance between two line segments (line 7 in Figure 4). One can bypass this computation by "filtering out" those line segments

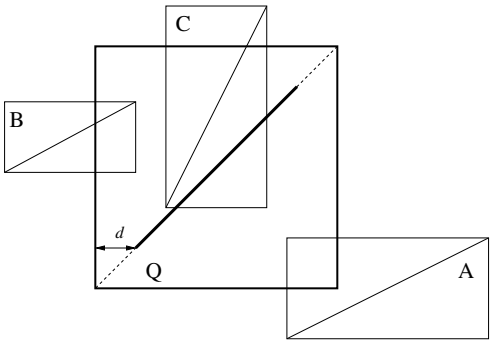


Figure 10. Three example entry MBBs and their overlap with a query MBB.

for which it is straightforward (i.e., computationally cheap) to determine that they cannot possibly lie within distance d of the query. This filtering is applied to the segments once they have been returned by the index, and is thus independent of the indexing method.

Figure 10 shows an example with a query MBB, Q, and three overlapping MBBs, A, B, and C, that have been returned from the index search. The query distance d is indicated in the

(augmented) query box so that the query trajectory segment is shorter than the box's diagonal. MBB A contains a segment that is outside Q and should thus be filtered out. The line segment in B crosses the query box boundary but is never within distance d of the query segment and should be filtered out. C contains a line segment that is within a distance d of the query segment, and should thus not be filtered out. For this segment a moving distance computation must be performed (Figure 4, line 7) to determine whether there is an interval of time in which the two trajectories are indeed within a distance d of each other. The fact that candidate segments are returned that should in fact be ignored is inherent to the use of MBBs: a segment occupies an infinitesimal portion of its MBB's space. This issue is germane to MODs that store trajectories using MBBs.

In practice, depending on the dataset and the search, the number of line segments that should be filtered out can be large. Figure 11 shows the number of candidate segments returned by the index search and the number of segments that are within the query distance vs. α, for the *Random*-1M dataset, with 100 randomly selected query trajectories processed for 100% of their temporal extent. The fraction of candidate segments that are within the query distance is below 16.5% at $\alpha = 1$. In this particular example, an ideal filtering method would filter out more than 80% of the line segments.

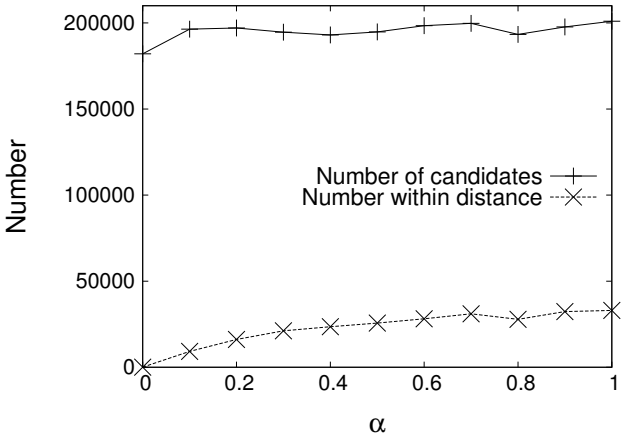


Figure 11. The number of moving distance calculations and the number that are actually within a distance of 15 vs. α in the *Random*-1M datasets.

A. Two Segment Filtering Methods

After the query and entry line segments are interpolated so that they have the same temporal extent (Figure 4, line 6), various criteria may remove the candidate segment from consideration. We consider two filtering methods beyond the simple no filtering approach:

- **Method 1 –** No filtering.
- **Method 2 –** After the interpolation, check whether the candidate segment still lies within the query MBB. This check only requires floating point comparisons between spatial coordinates of the segment endpoints and the query MBB corners, and would occur between lines 6 and 7 in Figure 4. Method 2 would filter out A in Figure 10.
- **Method 3 –** Considering only 2 spatial dimensions, say x and y, for a given query segment MBB compute the slope and the y-intercept of the line that contains the query segment. This computation requires only a few floating point operations and would occur in between lines 3 and 4 in Figure 4, i.e., in the outer loop. Then, before line 7, check if the endpoints of the candidate segment both lie more than a distance d above or below the query trajectory line. In this case, the candidate segment can be filtered out. This check requires only a few floating point operations involving segment endpoint coordinates and the computed slope and y-intercept of the query line. Method 3 would filter out both A and B in Figure 10.

Other computational geometry methods could be used for filtering, but these methods must be sufficiently fast (i.e., low floating point operation counts) if any benefit over Method 1 is to be achieved. For instance, one may consider a method that computes the shortest distance between an entry line segment and the query line segment regardless of time, and discard the candidate segment if this shortest distance is larger than threshold distance d. However, the number of (floating point) operations to perform such filtering is on the same order as that needed to perform the full-fledge moving distance calculation.

B. Filtering Performance

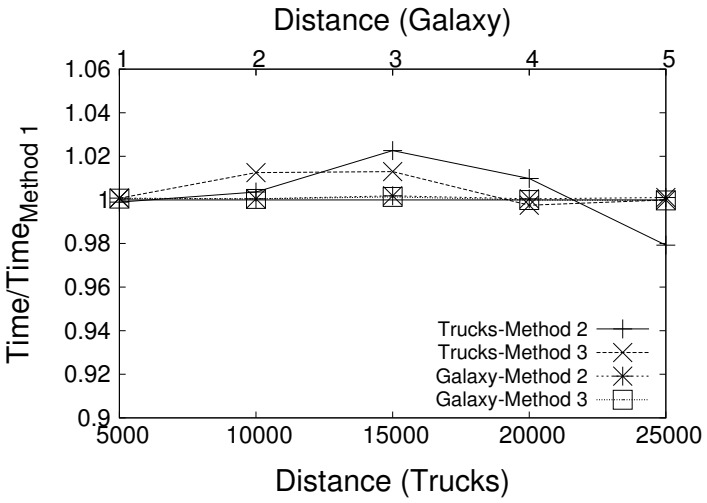
We have implemented the filtering methods in the previous section in TRAJDISTSEARCH and in this section we measure response times ignoring the R-tree search, i.e., focusing only on the filtering and the moving distance computation. We use the following distance threshold searches:

- S5: From the *Trucks* dataset, 10 trajectories are processed for 100% of their temporal extent.
- S6: From the *Galaxy*-1M dataset, 100 trajectories are processed for 100% of their temporal extent.
- S7: From the *Random*-1M datasets, 100 trajectories are processed for 100% of their temporal extent, with a fixed query distance $d = 15$.

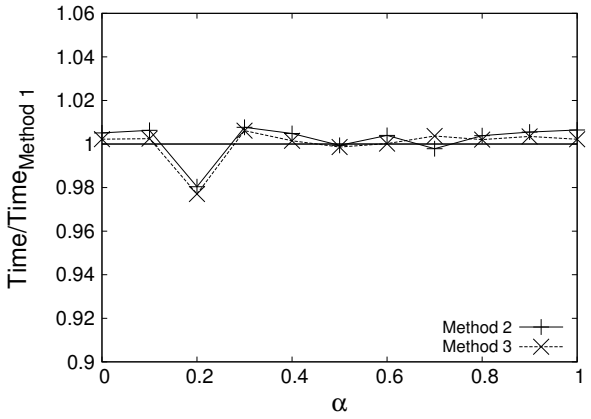
Figure 12 (a) plots the relative improvement (i.e., ratio of response times) of using Method 2 and Method 3 over using Method 1 vs. the threshold distance for S5 and S6 above for the Galaxy and Trucks datasets. Data points below the $y = 1$ line indicate that filtering is beneficial. We see that filtering is almost never beneficial and can in fact marginally increase response time. Similar results are shown for the *Random*-1M datasets in Figure 12 (b).

It turns out that our methods filter only a small fraction of the line segments. For instance, for search S7 Method 2, resp. Method 3, filters out between 2.5% and 12%, resp. between 3.2% and 15.9%, of the line segments. Therefore, for most candidate segments the time spent doing filtering is pure overhead. Furthermore, filtering requires only a few floating point operations but also several if-then-else statements. The resulting branch instructions slow down executions (due to pipeline stalls) when compared to straight line code. We conclude that, at least for the datasets and searches we have used, our filtering methods are not effective.

One may envision developing better filtering methods to achieve (part of the) filtering potential seen in Figure 11. We profiled the execution of TRAJDISTSEARCH for searches S5,



(a) *Trucks* and *Galaxy*-1M



(b) *Random*-1M

Figure 12. Performance improvement ratio of filtering methods (a) for real datasets with S5 and S6, vs. query distance, (b) for *Random*-1M datasets with S7.

S6, and S7, with no filtering, and accounting both for the R-tree search and the distance computation. We found that the time spent searching the R-tree accounts for at least 97% of the overall response time. As a result, filtering can only lead to marginal performance improvements for the datasets and searches in our experiments. For other datasets and searches, however, the fraction of time spent computing distances could be larger. Nevertheless, given the results in this section, in all that follows we do not perform any filtering.

VIII. Index Resolution

In this section, we propose methods to represent the trajectory segments in a different configuration within the index. According to the cost model in [35], index performance depends on the number of nodes in the index, but also on the volume and surface area of the MBBs. Query performance can be improved by finding a suitable number of nodes in the index combined with a good partitioning strategy of trajectory segments within MBBs. One extreme is to store an entire trajectory in a single MBB as defined by the spatial and temporal properties of the trajectory; however, this leads to a

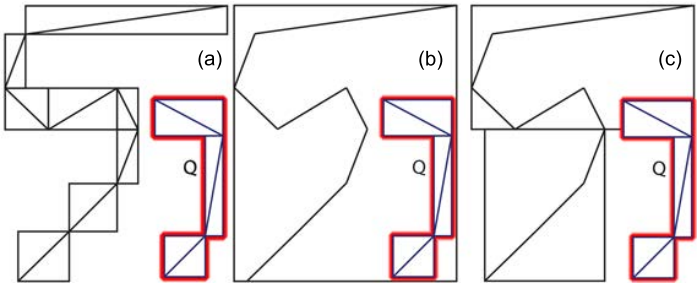


Figure 13. Illustration of the relationship between wasted space, volume occupied by indexed trajectories, and the number of returned candidate segments to process. An 8-segment trajectory is indexed in three different ways, and searched against a 3-segment query trajectory (denoted Q in the figure), where the query distance is shown in red. (a) Each trajectory segment is stored in its own MBB. (b) The trajectory is stored in a single MBB. (c) The trajectory is stored in two MBBs.

lot of "wasted MBB space." The other extreme is to store each trajectory line segment in its own MBB, as done so far in this paper and in previous work on kNN searches [22], [23], [24], [10]. In this scenario, the volume occupied by the trajectory in the index is minimized, with the trade-off that the number of entries in the index will be maximized.

In Figure 13 (a) we depict an entry trajectory that is stored with each segment in its own MBB, in Figure 13 (b), a trajectory that is stored in a single MBB, and in Figure 13 (c) a trajectory that is stored in two MBBs. A 3-segment query trajectory that is not within the query distance of the entry trajectory is shown, where the query distance is indicated by the red outline. Assigning a single line segment to a single MBB (Figure 13 (a)) minimizes wasted space but maximizes the number of nodes in the index that need to be searched. Storing an entire trajectory in its own MBB minimizes the number of index entries to be searched but leads to more index overlap and more candidate segments. For example, consider the query in Figure 13 (b). From the figure, it can be seen that each of the three query segments overlap the MBB, resulting in 3×8=24 candidate trajectory segments that need to be processed. However, in Figure 13 (a), the query trajectory does not overlap any of the entry MBBs, and therefore no candidate trajectory segments are returned; however, the index contains 8 elements instead of 1, as in Figure 13 (b). Figure 13 (c) shows the case in which the entry trajectory is stored in only 2 MBBs. In this case only 1 query segment overlaps an entry MBB, resulting in 1×5=5 candidate segments to process.

As shown above, assigning a fraction of a trajectory to a single MBB, as a series of line segments, increases the volume a trajectory occupies in the index, and the degree of index overlap. This is because the resulting MBB is larger in comparison to minimizing the volume of the MBBs by encompassing each individual trajectory line segment by its own MBB. As a result, an index search can return a portion of a trajectory that does not overlap the query, leading to increased overhead when processing the candidate set of line segments returned by the index. However, the number of entries in the index is reduced, thereby reducing tree traversal time. To explore the trade-off between the number of nodes in the index, the amount of wasted volume required by a trajectory, the index overlap, and the overhead of processing candidate

trajectory segments, in this section, we evaluate three strategies for splitting individual trajectories into a series of consecutive MBBs. Such splitting can be easily implemented as an array of references to trajectory segments (leading to one extra indirection when compared to assigning a single segment per MBB). We evaluate performance experimentally by splitting the trajectories, and then creating their associated indexes, where the configuration with the lowest query response time is highlighted.

A. Static Temporal Splitting

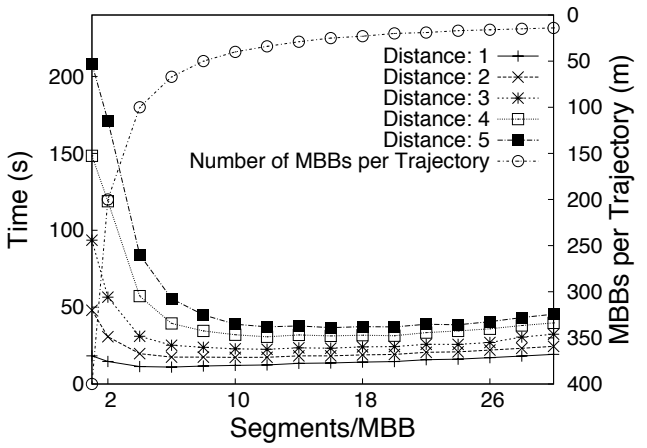
Assuming it is desirable to ensure that trajectory segments are stored contiguously, we propose a simple trajectory splitting method. Given a trajectory of n line segments, we split the trajectory by assigning r contiguous line segments per MBB, where r is a constant. Therefore, the number of MBBs, m, to represent the trajectory is $m = \lceil \frac{n}{r} \rceil$. By storing segments contiguously, this strategy leads to high temporal locality of reference, which may be important for cache reuse in our in-memory database, in addition to the benefits of the high spatial discrimination of the R-tree (see Section IV).

Figure 14 plots response time vs. r for the S6 (*Galaxy* dataset) and S7 (*Random* dataset) searches defined in Section VII-B. For S6, 5 different query distances are used, while for S7 the query distance is fixed as 15 but results are shown for various dataset sizes for $\alpha = 1$. The right y-axis shows the number of MBBs used per trajectory. The data points at $r = 1$ correspond to the original implementation (rather than the implementation with $r = 1$, which would include one unnecessary indirection).

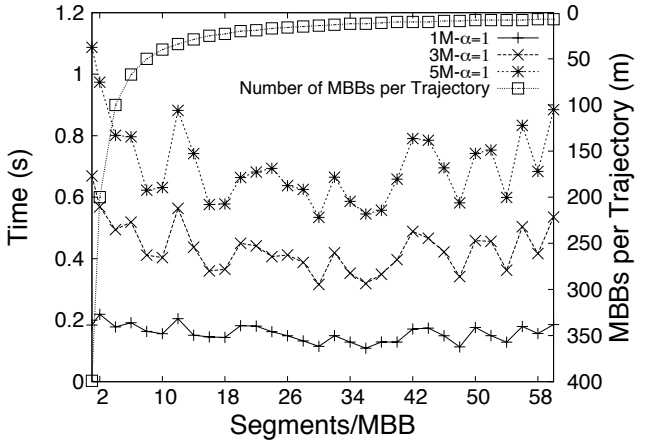
The best value for r depends on the dataset and the search. For instance, in the *Galaxy*-1M dataset (S6) using 12 segments per MBB (or $m = 34$) leads to the best performance. We note that picking a r value in a large neighborhood around this best value would lead to only marginally higher query response times. In general, using a small value of r can lead to high response times, especially for $r = 1$ (or $m = 400$). For instance, for S6 with a query distance of 5, the response time with $r = 1$ is above 208 s while it is just above 37 s with $r = 12$. With $r = 1$ the index is large and thus time-consuming to search. A very large r value does not lead to the lowest response time since in this case many of the segments returned from the R-tree search are not query matches. Finally, results in Figure 14 (a) show that the advantage of assigning multiple trajectory segments per MBB increases as the query distance increases. For instance, for a distance of 2 using $r = 12$ decreases the response time by a factor 2.76 when compared to using $r = 1$, while this factor is 5.6 for a distance of 5. Note that the difference in response times between Figure 14 (a) and (b) are largely due to more query hits in *Galaxy* in comparison to *Random* for the query distances selected.

B. Static Spatial Splitting

Another strategy consists in ordering the line segments belonging to a trajectory spatially, i.e., by sorting the line segments of a trajectory by the x, y, and z values of the segment's origin lexicographically. We then assign r segments per trajectory into each MBB, as in the previous method. With such spatial grouping, the line segments are no longer



(a) *Galaxy*-1M



(b) *Random* datasets

Figure 14. Static Temporal Splitting: Response time vs. r for (a) S6 for the *Galaxy*-1M dataset for various query distances; and (b) S7 for the *Random*-1M, 3M, and 5M $\alpha = 1$ datasets and a query distance of 15. The number of MBBs per trajectory, m, is shown on the right vertical axis.

guaranteed to be temporally contiguous in their MBBs, but reduced index overlap may be achieved. Figure 15 plots response time vs. r for the S7 (*Random* dataset) searches. We see that there is no advantage to assigning multiple trajectory segments to an MBB over assigning a single line segment to a MBB ($r = 1$ in the plot). When comparing with results in Figure 14 (b) we find that spatial splitting leads to query response times higher by several factors than that of temporal splitting.

C. Splitting to Reduce Trajectory Volume

The encouraging results in Section VIII-A suggest that using an appropriate trajectory splitting strategy can lead to performance gains primarily by exploiting the trade-off between the number of entries in the index and the amount of wasted space that leads to higher index overlap. More sophisticated methods can be used. In particular, we implement the heuristic algorithm *MergeSplit* in [36], which is shown to produce a splitting close to optimal in terms of wasted space. *MergeSplit* takes as input a trajectory, T, as a series of l line segments, and a constant number of MBBs, m. As output, the algorithm creates a set of m MBBs that encapsulate the l segments of T. The pseudocode of *MergeSplit* is as follows:

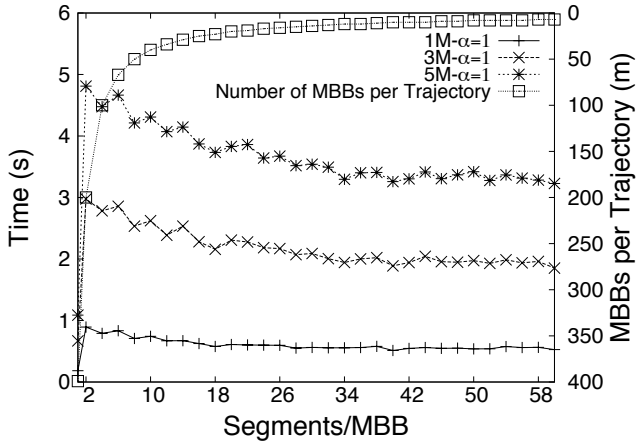


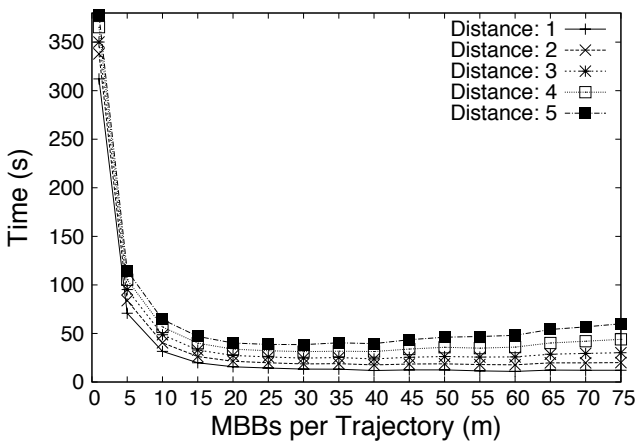
Figure 15. Static Spatial Splitting: Response time vs. r using S7 for the *Random*-1M, 3M, and 5M $\alpha = 1$ datasets and a query distance of 15. The number of MBBs per trajectory, m, for each data point is shown on the rightmost vertical axis.

1) For $0 \leq i < l$ calculate the volume of the merger of the MBBs that define l_i and l_{i+1} and store the resulting series of MBBs and their volumes.
2) To obtain m MBBs, merge consecutive MBBs that produce the smallest volume increase at each iteration and repeat $(l-1)-(m-1)$ times. After the first iteration, there will be $l-2$ initial MBBs describing line segments, and one MBB that is the merger of two line segment MBBs.

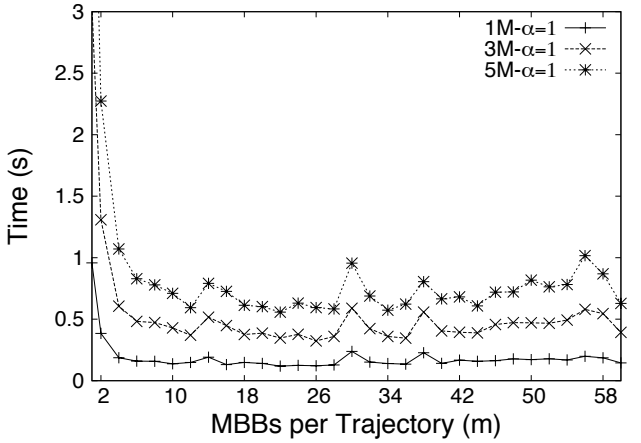
Figure 16 shows response time vs. m for S6 (*Galaxy* dataset) and S7 (*Random* datasets). Compared to static temporal splitting, which has a constant number of segments, r per MBB, *MergeSplit* has a variable number of segments per MBB. From the figure, we observe that for the *Galaxy*-1M dataset (S6), $m = 30$ leads to the best performance. Comparing *MergeSplit* to the static temporal splitting (Figures 14 and 16 (a)), the best performance for the S6 (*Galaxy* dataset) is achieved by the static temporal splitting. For S7, the *Random*-1M, 3M, and 5M $\alpha = 1$ datasets, *MergeSplit* is only marginally better than the static temporal splitting (Figures 14 and 16 (b)). This is surprising, given that the total hypervolume of the entries in the index for a given m across both splitting strategies is higher for the simple static temporal splitting, as it makes no attempt to minimize volume. Therefore, the trade-off between the number of entries and overlap in the index cannot fully explain the performance of these trajectory splitting strategies for distance threshold searches.

D. Discussion

A good trade-off between the number of entries in the index and the amount of index overlap can be achieved by selecting an appropriate trajectory splitting strategy. However, comparing the results of the simple temporal splitting strategy (Section VIII-A) and *MergeSplit* (Section VIII-C), we find that volume minimization did not significantly improve performance for S7, and led to worse performance for S6. In Figure 17, we plot the total hypervolume vs. m for the *Galaxy*-1M (S6) and the *Random*-1M, 3M, and 5M $\alpha = 1$ (S7) datasets. $m = 1$ refers to placing an entire trajectory in a



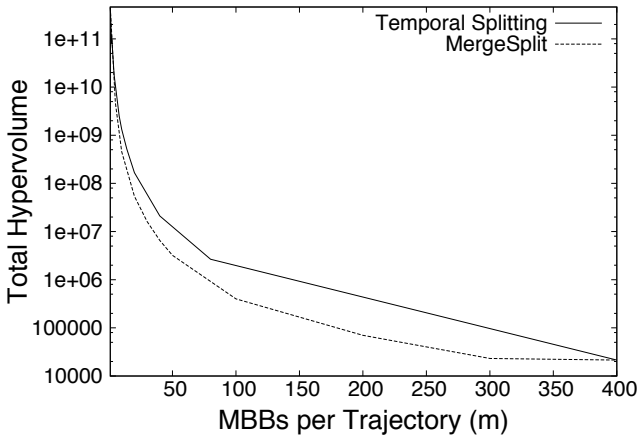
(a) *Galaxy*-1M



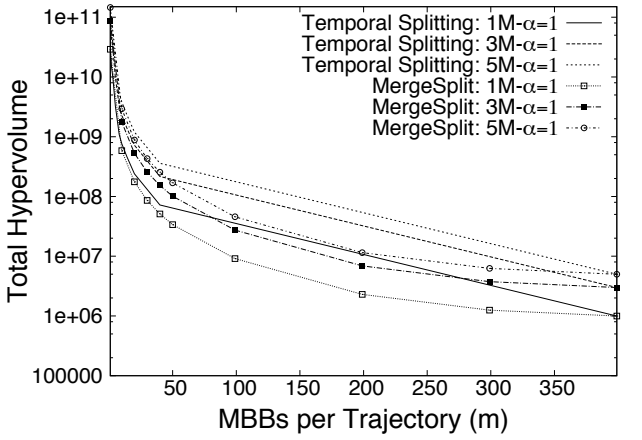
(b) *Random* datasets

Figure 16. Greedy Trajectory Splitting: Response time vs. m for (a) S6 for the *Galaxy*-1M dataset for various query distances; and (b) S7 for the *Random*-1M, 3M, and 5M $\alpha = 1$ datasets and a query distance of 15.

single MBB, and the maximum value of m refers to placing each individual line segment of a trajectory in its own MBB. For the static temporal splitting strategy, $m = 34$ leads to the best performance for the *Galaxy*-1M dataset (S6), whereas this value is $m = 30$ for *MergeSplit*. The total hypervolume of the MBBs in units of kpc^{3}Gyr for the static temporal grouping strategy at $m = 34$ is 3.6×10^7, whereas for *MergeSplit* at $m = 30$, it is 1.62×10^7, i.e., 55% less volume. Due to the greater volume occupied by the MBBs, index overlap is much higher for the static temporal splitting strategy. Figure 18 (a) plots the number of overlapping line segments vs. m for S6 with $d = 5$. From the figure, we observe that independently of m, *MergeSplit* returns a greater number of candidate line segments to process than the simple temporal splitting strategy. *MergeSplit* attempts to minimize volume; however, if an MBB contains a significant fraction of the line segments of a given trajectory, then all of these segments are returned as candidates. The simple temporal grouping strategy has an upper bound (r) on the number of segments returned per overlapping MBB and thus can return fewer candidate segments for a query, despite occupying more volume in the index. For in-memory distance threshold searches, there is a trade-off between a trajectory splitting strategy that has an upper bound on the number of line segments per MBB, and index overlap, characterized

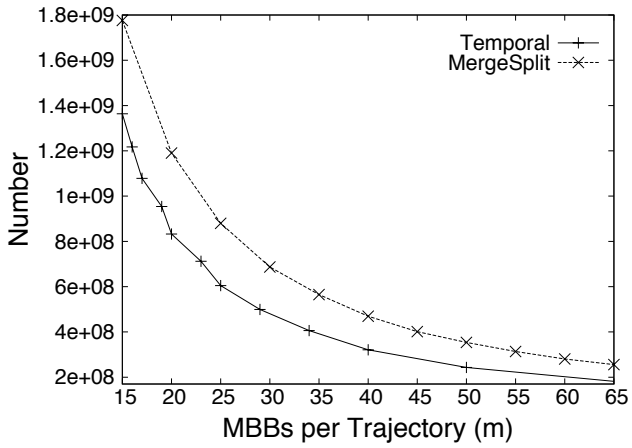


(a) *Galaxy*-1M

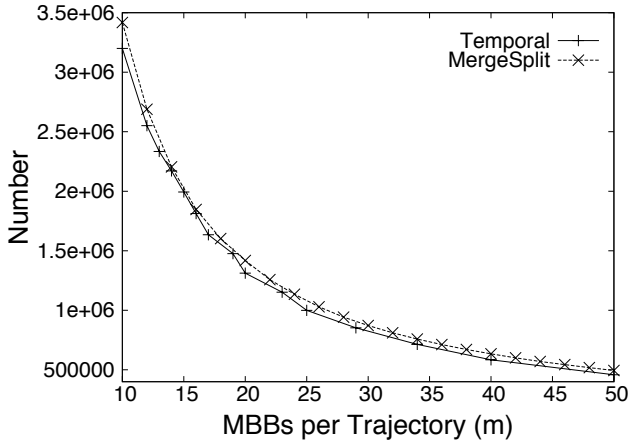


(b) *Random*-1M

Figure 17. Total hypervolume vs. m for the static temporal splitting strategy and *MergeSplit*. (a)for the *Galaxy*-1M dataset (S6); and (b) for the *Random*-1M, 3M, and 5M $\alpha = 1$ datasets (S7).



(a) *Galaxy*-1M



(b) *Random*-1M

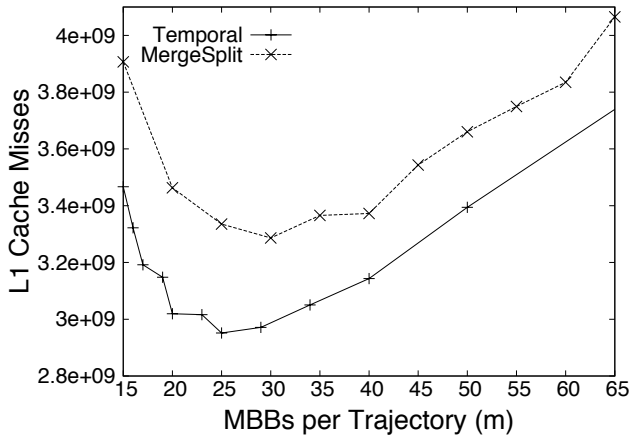
Figure 18. Total number of overlapping segments vs. m for the static temporal splitting strategy and *MergeSplit*. (a) S6 for the *Galaxy*-1M dataset with $d = 5$; and (b) S7 for the *Random* $\alpha = 1$ dataset with $d = 15$.

by the volume occupied by the MBBs in the index. This is in sharp contrast to other works that focus on efficient indexing of spatiotemporal objects in traditional out-of-core implementations where the index resides partially in-memory and on disk, and therefore volume reduction to minimize index overlap is necessary to minimize disk accesses (e.g., [36]).

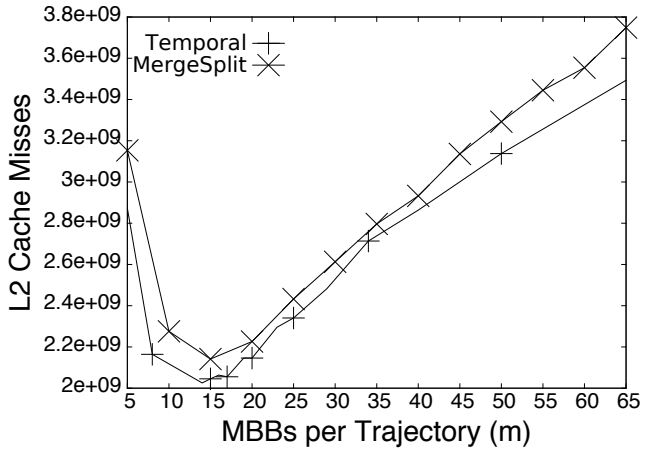
A single metric cannot capture the trade-offs between the number of entries in the index, volume reduction, index overlap, and the number of candidate line segments returned (germane to distance threshold searches). However, for *Galaxy*-1M (S6), a value of $m = 34$ and $m = 30$ lead to the best query response time for the temporal splitting strategy and *MergeSplit*, respectively (Figures 14 (a) and 16 (a)). Figure 19 (a) shows the number of L1 cache misses vs. m for S6 with $d = 5$. The number of cache misses was measured using PAPI [37]. The best values of m in terms of query response time for both of the trajectory splitting strategies ($m = 34$ and $m = 30$) roughly correspond to a value of m that minimizes L1 cache misses. Thus, L1 cache misses appear to be a good indicator of relative query performance under different indexing methods. Figure 19 (b) shows the number L2 cache misses vs. m for S6 with $d = 5$. We note that when comparing Figure 19 (a) and (b), there are more L1 cache misses for a given value of m because the L1 cache is smaller than L2 cache. We see that unlike L1 cache misses, m values that minimize L2 cache misses do not lead to the best response times for either splitting strategy. Therefore, L1 cache misses are a better predictor of query performance when comparing indexing methods. Future work for in-memory distance threshold searches should focus on improved cache reuse through temporal locality of reference (which is in part obtained by storing segments contiguously within a single MBB).

E. Performance Considerations for In-memory vs. Out-of-Core Implementations

The focus of this work is on in-memory distance threshold searches; however, most of the literature on MODs assume out-of-core implementations, where the number of node accesses are used as a metric to estimate I/O activity. Figure 20 shows the number of node accesses vs. m for both of the static temporal splitting strategy and *MergeSplit*. We find that for the *Galaxy*-1M dataset (S6) with $d = 5$, there is a comparable number of node accesses for both trajectory splitting methods. However, for S7 (*Random*-1M), on average, trajectory splitting with *MergeSplit* requires fewer node accesses and may

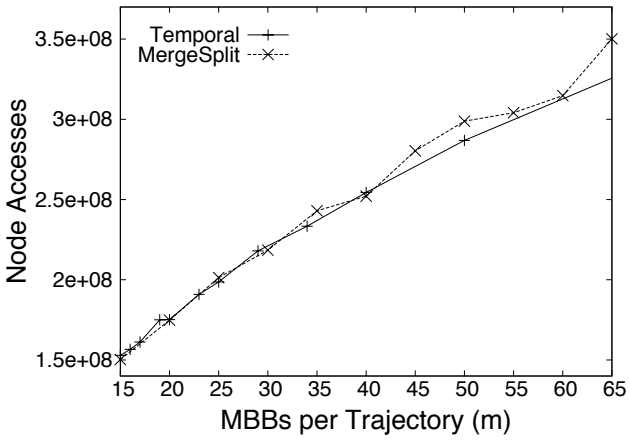


(a) *Galaxy*-1M

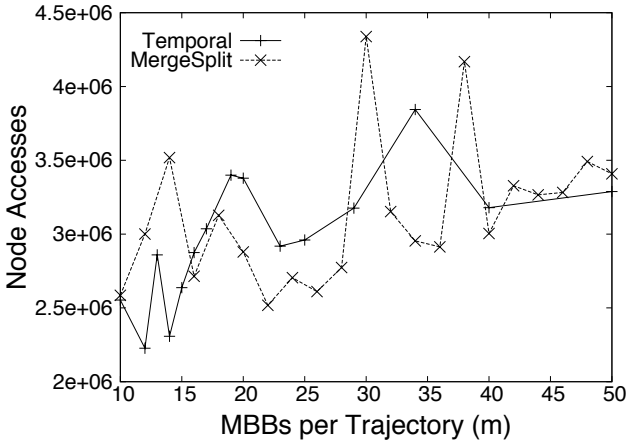


(b) *Random*-1M

Figure 19. L1 (a) and L2 (b) cache misses vs. m for the static temporal splitting strategy and *MergeSplit* for the *Galaxy*-1M dataset (S6) with $d = 5$.



(a) *Galaxy*-1M



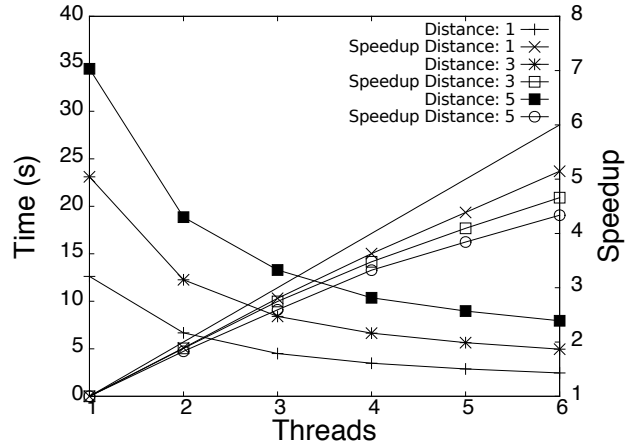
(b) *Random*-1M

Figure 20. Node Accesses vs. m for the static temporal splitting strategy and *MergeSplit*. (a) S6 for the *Galaxy*-1M dataset with $d = 5$; and (b) S7 for the *Random* $\alpha = 1$ dataset with $d = 15$.

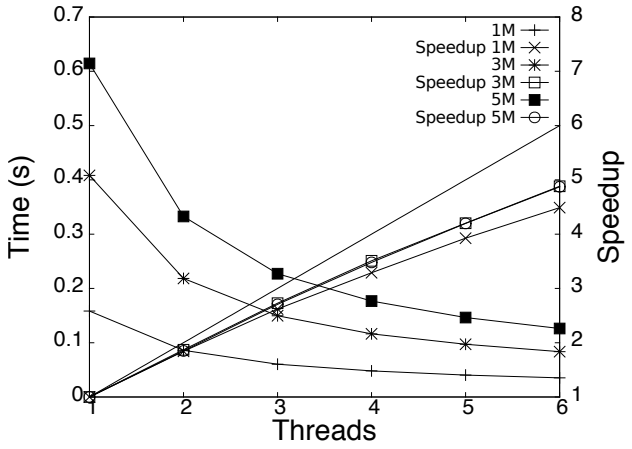
perform significantly better than the simple temporal splitting strategy in an out-of-core implementation. For example, in Figure 20 (b) some values of m have a significantly higher number of node accesses, such as values around 14, 30, 38, due to the idiosyncrasies of the data, and resulting index overlap. However, as we demonstrated in Section VIII-D, distance threshold searches in the context of in-memory databases also benefit from reducing the number of candidate line segments returned, and this is not entirely volume contingent. Therefore, methods that consider volume reduction, such as the *Merge-Split* algorithm of [36], or other works that consider volume reduction in the context of query sizes, such as [38], may not be entirely applicable to distance threshold searches.

F. Multi-core Execution with OpenMP

In Section VIII-D, we noted that indexing multiple line segments in a single MBB leads to performance improvements and that the temporal splitting strategy performed better than the spatial splitting strategy and *MergeSplit*. Regardless of the trajectory splitting strategy utilized, TRAJDISTSEARCH can be parallelized, e.g., using OpenMP, in a shared-memory environment. The iterations of the loop on line 3 of TRAJDISTSEARCH in Figure 4 are independent, each iteration can thus be assigned to a different thread. Figure 21 shows the response time on the 6-core platform described in Section VI-B vs. the number of threads for S6 and S7 with $r = 12$ and $r = 10$, respectively. These values of r yield the best performance gain in the sequential implementation for S6 and S7 (Figure 14). Parallelizing the outer loop leads to high parallel efficiency between 72.2%-85.7%, with parallel speedup between 4.33 and 5.14 with 6 threads for query distances ranging from $d = 1$ to $d = 5$ for the *Galaxy* dataset with S6. For the *Random*-1M, 3M and 5M $\alpha = 1$ datasets, with 6 threads, we observe a speedup between 4.49 to 4.88, for a parallel efficiency between 74.8% and 81.3%. We note from Figure 21 (a) that the speedup decreases as d increases. This suggests that as the number of candidate segments increases (with increasing d), there is likely to be increased memory contention, as more candidate segments between the threads are competing for space in the CPU cache. Additionally, with an increased d, there will be more nodes to visit in the R-tree; however, the threads can traverse the tree in parallel. It is not clear which mechanism predominantly causes the slowdown with increasing query distance. However, from previous sections we saw that the number of candidate trajectory segments can be large, and it is likely that processing candidate trajectory segments is the main bottleneck in parallelizing distance threshold searches.



(a) *Galaxy*



(b) *Random* datasets

Figure 21. Response time vs. number of threads (a) S6 for the *Galaxy* dataset for various query distances and $r = 12$; and (b) S7 for the *Random*-1M, 3M, and 5M datasets, with a query distance of 15 and $r = 10$.

Distance threshold searches, and perhaps other spatiotemporal searches on trajectories can be parallelized in a straightforward manner in a shared-memory environment because the searches can be performed independently of each other. The focus in the spatiotemporal database community has been on out-of-core, sequential implementations; however, with new architectures, and large main memories, there are a number of attractive alternatives to the current solutions.

IX. Conclusion

In-memory distance threshold searches for trajectory and point searches on moving object trajectories are significantly different from the well-studied kNN searches [22], [23], [24], [10]. We made a case for using an R-tree index to store trajectory segments, and found it to perform robustly for two real world datasets and a synthetic dataset. We focused on 4-D datasets (3 spatial + 1 temporal) while other works only consider 3-D datasets [22], [23], [24], [10].

We demonstrated that for distance threshold searches, many segments returned by the index search must be excluded from the result set, because there is no limit to the number of candidate trajectory segments that can be returned. We have proposed computationally inexpensive solutions to filter out candidate segments, but found that they have poor selectivity. A more promising direction for reducing query response time is to reduce the time spent traversing the tree index.

We demonstrated that efficiently splitting trajectories is beneficial because the penalty for the increased index overlap is offset by the reduction in number of index entries. We find that for in-memory distance threshold searches, the number of line segments returned per overlapping MBB has an impact on performance, where attempts to reduce the volume of the MBBs that store a trajectory may be at cross-purposes with returning a limited number of candidate segments per overlapping MBB. Therefore, at least for in-memory implementations, trajectory splitting methods that focus on volume reduction are not necessarily preferable to a simple and bounded grouping of line segments in MBBs for distance threshold searches.

We show that the distance threshold search can be performed in parallel using threads in a shared-memory environment using OpenMP. The results show that the tree traversals and processing of candidate segments can be performed in parallel with high parallel efficiency. The results are encouraging for future prospects in parallel query optimization, and suggests that a promising future work direction is to investigate both shared- and distributed-memory implementations.

A future direction is to explore trajectory splitting methods that achieve volume reduction while bounding the number of MBBs used per trajectory. Another direction is to investigate non-MBB-based data structures to index line segments, such as that in [39]. Analytical models of query performance in these settings may be heavily dependent on modeling cache reuse.

One may wonder whether the idea of assigning multiple segments to an MBB is generally applicable, and in particular for kNN searches on trajectories [22], [23], [24], [10]. The kNN literature focuses on pruning strategies and associated metrics that require a high resolution index, thus implying storing a single trajectory segment in an MBB. Furthermore, kNN algorithms maintain a list of nearest neighbors over a time interval, which would lead to greater overhead if multiple segments were stored per MBB. Therefore, the approach of grouping line segments together in a single MBB may be ineffective for kNN searches. An interesting problem is to reconcile the differences between both types of searches in terms of index resolution.

Acknowledgments

This material is based upon work supported by the National Aeronautics and Space Administration through the NASA Astrobiology Institute under Cooperative Agreement No. NNA08DA77A issued through the Office of Space Science, and by NSF Award CNS-0855245.

References

[1] M. Gowanlock and H. Casanova, "In-memory distance threshold queries on moving object trajectories," in Proc. of the Sixth Intl. Conf. on Advances in Databases, Knowledge, and Data Applications, 2014, pp. 41–50.

[2] L. Forlizzi, R. H. Güting, E. Nardelli, and M. Schneider, "A data model and data structures for moving objects databases," in Proc. of ACM SIGMOD Intl. Conf. on Management of Data, 2000, pp. 319–330.

[3] R. H. Güting, M. H. Böhlen, M. Erwig, C. S. Jensen, N. A. Lorentzos, M. Schneider, and M. Vazirgiannis, "A foundation for representing and querying moving objects," ACM Trans. Database Syst., vol. 25, no. 1, 2000, pp. 1–42.

[4] S. Arumugam and C. Jermaine, "Closest-point-of-approach join for moving object histories," in Proc. of the 22nd Intl. Conf. on Data Engineering, 2006, pp. 86–95.

[5] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of convoys in trajectory databases," Proc. VLDB Endow., vol. 1, no. 1, Aug. 2008, pp. 1068–1080.

[6] A. Guttman, "R-trees: a dynamic index structure for spatial searching," in Proc. of ACM SIGMOD Intl. Conf. on Management of Data, 1984, pp. 47–57.

[7] D. Pfoser, C. S. Jensen, and Y. Theodoridis, "Novel approaches in query proc. for moving object trajectories," in Proc. of the 26th Intl. Conf. on Very Large Data Bases, 2000, pp. 395–406.

[8] Y. Theodoridis, M. Vazirgiannis, and T. Sellis, "Spatio-temporal indexing for large multimedia applications," in Proc. of the Intl. Conf. on Multimedia Computing and Systems, 1996, pp. 441–448.

[9] V. P. Chakka, A. Everspaugh, and J. M. Patel, "Indexing large trajectory data sets with SETI," in Proc. of Conference on Innovative Data Systems Research, 2003, pp. 164–175.

[10] R. H. Güting, T. Behr, and J. Xu, "Efficient k-nearest neighbor search on moving object trajectories," The VLDB Journal, vol. 19, no. 5, 2010, pp. 687–714.

[11] P. Cudre-Mauroux, E. Wu, and S. Madden, "TrajStore: an adaptive storage system for very large trajectory data sets," in Proc. of the 26th Intl. Conf. on Data Engineering, 2010, pp. 109–120.

[12] M. R. Vieira, P. Bakalov, and V. J. Tsotras, "On-line discovery of flock patterns in spatio-temporal data," in Proc. of the 17th ACM SIGSPATIAL Intl. Conf. on Advances in Geographic Information Systems, 2009, pp. 286–295.

[13] Z. Li, M. Ji, J.-G. Lee, L.-A. Tang, Y. Yu, J. Han, and R. Kays, "Movemine: Mining moving object databases," in Proc. of the 2010 ACM SIGMOD Intl. Conf. on Management of Data, 2010, pp. 1203–1206.

[14] N. Roussopoulos, S. Kelley, and F. Vincent, "Nearest neighbor queries," in Proc. of ACM SIGMOD Intl. Conf. on Management of Data, 1995, pp. 71–79.

[15] Z. Song and N. Roussopoulos, "K-nearest neighbor search for moving query point," in Proc. of the 7th Intl. Symp. on Advances in Spatial and Temporal Databases, 2001, pp. 79–96.

[16] Y. Tao, D. Papadias, and Q. Shen, "Continuous nearest neighbor search," in Proc. of the 28th Intl. Conf. on Very Large Data Bases, 2002, pp. 287–298.

[17] R. Benetis, S. Jensen, G. Karciauskas, and S. Saltenis, "Nearest and reverse nearest neighbor queries for moving objects," The VLDB Journal, vol. 15, no. 3, 2006, pp. 229–249.

[18] K. Mouratidis, D. Papadias, and M. Hadjieleftheriou, "Conceptual partitioning: an efficient method for continuous nearest neighbor monitoring," in Proc. of ACM SIGMOD Intl. Conf. on Danagement of data, 2005, pp. 634–645.

[19] K. Mouratidis, D. Papadias, S. Bakiras, and Y. Tao, "A threshold-based algorithm for continuous monitoring of k nearest neighbors," IEEE Trans. on Knowl. and Data Eng., vol. 17, no. 11, 2005, pp. 1451–1464.

[20] X. Xiong, M. F. Mokbel, and W. G. Aref, "SEA-CNN: Scalable proc. of continuous k-nearest neighbor queries in spatio-temporal databases," in Proc. of the 21st Intl. Conf. on Data Engineering, 2005, pp. 643–654.

[21] X. Yu, K. Q. Pu, and N. Koudas, "Monitoring k-nearest neighbor queries over moving objects," in Proc. of the 21st Intl. Conf. on Data Engineering, 2005, pp. 631–642.

[22] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis, "Nearest neighbor search on moving object trajectories," in Proc. of the 9th Intl. Conf. on Advances in Spatial and Temporal Databases, 2005, pp. 328–345.

[23] E. Frentzos, K. Gratsias, N. Pelekis, and Y. Theodoridis, "Algorithms for nearest neighbor search on moving object trajectories," Geoinformatica, vol. 11, no. 2, 2007, pp. 159–193.

[24] Y.-J. Gao, C. Li, G.-C. Chen, L. Chen, X.-T. Jiang, and C. Chen, "Efficient k-nearest-neighbor search algorithms for historical moving object trajectories," J. Comput. Sci. Technol., vol. 22, no. 2, 2007, pp. 232–244.

[25] S. Qiao, C. Tang, S. Dai, M. Zhu, J. Peng, H. Li, and Y. Ku, "Partspan: Parallel sequence mining of trajectory patterns," in Fifth Intl. Conf. on Fuzzy Systems and Knowledge Discovery, vol. 5, Oct 2008, pp. 363–367.

[26] R. Jinno, K. Seki, and K. Uehara, "Parallel distributed trajectory pattern mining using mapreduce," in 2012 IEEE 4th Intl. Conf. on Cloud Computing Technology and Science (CloudCom), Dec 2012, pp. 269–273.

[27] F. Giannotti, M. Nanni, F. Pinelli, and D. Pedreschi, "Trajectory pattern mining," in Proc. of the 13th ACM SIGKDD Intl. Conf. on Knowledge Discovery and Data Mining, 2007, pp. 330–339.

[28] D. Zeinalipour-Yazti, S. Lin, and D. Gunopulos, "Distributed spatio-temporal similarity search," in Proc. of the 15th ACM Intl. Conf. on Information and Knowledge Management. ACM, 2006, pp. 14–23.

[29] J. Zhang, S. You, and L. Gruenwald, "Parallel online spatial and temporal aggregations on multi-core CPUs and many-core GPUs." Information Systems, vol. 44, 2014, pp. 134 – 154.

[30] M. G. Gowanlock, D. R. Patton, and S. M. McConnell, "A model of habitability within the Milky Way galaxy," Astrobiology, vol. 11, 2011, pp. 855–873.

[31] http://www.chorochronos.org/, accessed 5-February-2014.

[32] Y. Theodoridis, J. R. O. Silva, and M. A. Nascimento, "On the generation of spatiotemporal datasets," in Proc. of the 6th Intl. Symp. on Advances in Spatial Databases, 1999, pp. 147–164.

[33] http://navet.ics.hawaii.edu/%7Emike/datasets/DBKDA2014/datasets.zip, accessed 12-February-2014.

[34] http://www.superliminal.com/sources/sources.htm, accessed 5-February-2014.

[35] B.-U. Pagel, H.-W. Six, H. Toben, and P. Widmayer, "Towards an analysis of range query performance in spatial data structures," in Proc. of the 12th Symp. on Principles of Database Sys., 1993, pp. 214–221.

[36] M. Hadjieleftheriou, G. Kollios, V. J. Tsotras, and D. Gunopulos, "Efficient indexing of spatiotemporal objects," in Proc. of the 8th Intl. Conf. on Extending Database Technology: Advances in Database Technology, 2002, pp. 251–268.

[37] P. J. Mucci, S. Browne, C. Deane, and G. Ho, "PAPI: A portable interface to hardware performance counters," in Proc. of the Department of Defense HPCMP Users Group Conf., 1999, pp. 7–10.

[38] S. Rasetic, J. Sander, J. Elding, and M. A. Nascimento, "A trajectory splitting model for efficient spatio-temporal indexing," in Proc. of the 31st Intl. Conf. on Very Large Data Bases, 2005, pp. 934–945.

[39] E. Bertino, B. Catania, and B. Shidlovsky, "Towards optimal indexing for segment databases," in Proc. of the 6th Intl. Conf. on Advances in Database Technology, 1998, pp. 39–53.

Automated Feature Construction for Classification of Time Ordered Data Sequences

Michael Schaidnagel, Thomas Connolly
School of Computing
University of the West of Scotland
Email: B00260359@studentmail.uws.ac.uk
Thomas.Connolly@uws.ac.uk

Fritz Laux
Faculty of Computer Science
Reutlingen University
Email: fritz.laux@reutlingen-university.de

***Abstract*—The recent years and especially the Internet have changed the ways in which data is stored. It is now common to store data in the form of transactions, together with its creation time-stamp. These transactions can often be attributed to logical units, e.g., all transactions that belong to one customer. These groups, we refer to them as data sequences, have a more complex structure than tuple-based data. This makes it more difficult to find discriminatory patterns for classification purposes. However, the complex structure potentially enables us to track behaviour and its change over the course of time. This is quite interesting, especially in the e-commerce area, in which classification of a sequence of customer actions is still a challenging task for data miners. However, before standard algorithms such as Decision Trees, Neural Nets, Naive Bayes or Bayesian Belief Networks can be applied on sequential data, preparations are required in order to capture the information stored within the sequences. Therefore, this work presents a systematic approach on how to reveal sequence patterns among data and how to construct powerful features out of the primitive sequence attributes. This is achieved by sequence aggregation and the incorporation of time dimension into the feature construction step. The proposed algorithm is described in detail and applied on a real-life data set, which demonstrates the ability of the proposed algorithm to boost the classification performance of well-known data mining algorithms for binary classification tasks.**

***Index Terms*—feature construction, sequential data, temporal data mining**

I. Introduction

This work extends our previous work in the field of feature construction reported in [1]. It presents new feature construction techniques as well as new experimental results.

Significant amounts of data are being generated on a daily basis, in almost every industry and scientific research area. Advancements in computer science as well as computer hardware enable us to store these data. The rate of growth of data surpasses the capability of analysing all the stored data. It is believed that less than 10 % of all data stored is retrieved or analysed [2]. Particularly in the e-commerce area it is common to log all user activities in an online shop. Such data can be ordered by their timestamp and can be allocated to data sequences of particular users. However, the logged activities or actions are not stored in a form that enables immediate data mining. Therefore, it is important to pre-process the data before analyzing it (see also [3] [4]). When data is only represented by primitive attributes and there is no prior domain expert knowledge available, the pre-processing task becomes challenging and creates the need for automated techniques. At this point attribute selection and/or feature construction techniques need to be applied. Attribute selection can be defined as the task of selecting a subset of attributes, which are able to perform at least as good on a given data mining task as the original attributes set. The original values of the data set are called attributes, while the constructed data are called features. It is possible that primitive attributes are not able to adequately describe eventually existing relations among primitive attributes. Such interrelations (or also called interactions [6]) can occur in a data set if the relation between one attribute and the target concept depends on another attribute (see also [7]).

A. Structure of the paper

The remainder of this section will provide a short introduction into the related fields of Feature Construction and Sequential Data Mining. It will also present the problem at hand. Section II will provide a short overview about the related research fields and briefly introduces well-known Feature Construction techniques. Subsection II-C will highlight our contribution to the particular research field of sequential data classification. The characteristics of such data are described in Section III. Our approach to sequential feature construction will be described in detail in Section IV. This is followed by an experimental analysis in Section V, in which we demonstrate the ability of our proposed algorithm to boost classification performance on a real-life data set. The paper then provides some conclusion (Section VI) and future work (Section VII).

B. Feature Construction

Attribute selection alone can fail to find existing interaction among data. Therefore, one goal for feature construction is to find and highlight interactions. Feature construction can be defined as the process of creating new compound properties using functional expressions on the primitive attributes. There are also the terms Attribute Construction (Han and Kamber [3]) and Feature Extraction (Guyon et al. [9]) used in the literature to denote this research area. Guyon et al. are more focused on the Feature Selection task and uses the term Feature

Extraction as a compound term to denote both Feature Construction and Feature Selection tasks. This work will continue to use the term Feature Construction. Feature Construction is part of the data preparation step within the KDD process. There are two groups of data preparation techniques: one contains techniques that do not alter the space dimensionality of the given data, such as signal enhancement, normalization and standardization. The other group reduces or enlarges the feature space. Examples for this group are non-linear expansions, feature discretization [9]. Feature Construction can fit into both groups, depending on the goals of the data preparation step. Another goal of Feature Construction is to reduce the data dimension by removing redundant or irrelevant attributes [7]. This is done by constructing new features out of several of the given attributes to help the mining process [3]. In this case the constructed feature replaces the attributes it was constructed from [7]. However, it is important to not discard important information, which is necessary to describe the target hypothesis. If done correctly, Feature Construction is the key data preparation step to build classifiers that are able to describe complex patterns. The positive impact of feature construction was also shown in a comparative study focusing on predictive accuracy [7]. Liu and Motoda define Feature Construction as the process *'that discovers missing information about the relationships between features and augments the space of features by inferring or creating additional features'* [4]. This means that the original representation of data is altered and the feature space is extended by the new features. Usually, logical operators are used to combine features. A simple example for a Feature Construction technique on a two dimensional problem is the following: assume that A_1 is the width and A_2 is the length of a rectangle. This can be transformed into a one-dimensional problem by creating the feature F_1 as area $F_1 = A_1 * A_2$ [4]. However, the success of feature construction is dependent on the goal of the Data Mining problem at hand. There is no use in calculating the area as a feature; if the pattern (that can be used for discrimination between the two given labels) is connected to the aspect ratio of the rectangles. Shafti and Pérez distinguish between two types of features construction techniques in terms of their construction strategy:

- hypothesis-driven: create features based on a hypothesis (which is expressed as a set of rules). These features are then added to the original data set and are used for the next iteration, in which a new hypothesis will be tested. This process continues until a termination requirement is satisfied.
- data-driven methods: create features based on predetermined functional expressions, which are applied on combinations of primitive features of a data set. These strategies are normally non-iterative and the new features are evaluated by directly assessing the data.

The transformation of the feature space is a standard procedure in Data Mining, since it may improve the recognition process of classifiers. In general the transformation function is denoted as $y = F(x)$. It is used to transform an n-dimensional original pattern x, that exists as a vector of the n-dimensional pattern space, into an m-dimensional pattern y [10]. Finding a good transformation function is very domain specific and also depends on the available measurements [9]. After the transformation, data objects are represented as feature vectors in the expanded and augmented feature space. This effectively pulls apart examples of the same class, so that it is easier for the classifier to distinguish them [12]. However, Feature Construction must be used with some precautions: if a new classification problem is presented, it is not obvious, which of the various data representations should be used. It is also possible that none of the constructed features are able to express the target concept sufficiently. This can be the case in many real-life scenarios and it needs to be dealt with by using domain-specific knowledge. Feature Construction techniques are mostly based on a fixed set of basic operators. There is no easy way to alter the existing constructor set. This is also a disadvantage if the classification problem requires a combination of several construction functions to find a discriminatory form of data representation [11]. Some feature construction techniques only use Boolean representations of features, which cover only part of the potential relations between data attributes. Markovitch and Rosenstein [11] also points out that the basic Boolean operators such as AND and OR are already inherently represented in the structure of a decision tree.

C. Sequential Data Mining

Feature Construction prepares the data before the actual mining is done. This can be difficult, if the data has a complex structure, such as sequences. Therefore, this section briefly introduces the rather young research field of Sequential Data Mining. Due to the increasing ability to store complex data sequences, it has become one of the most important and active subfields of data mining research. Dong and Pei [19] define it as special subfield of data mining for certain structured data. The term structured data thereby refers to data that is structured in an explicit way and comprises of a set of data items. In terms of sequences this structure can include (partial) orderings, temporal orderings, hierarchical structure as well as network structures. A more formal description of the sequence structure is also given in Section III. Other forms of structured data, which are not in the scope of this work, are for example tree data, graph data, time series data or also text data.

The complex data structure of sequences is what sets Sequential Data Mining apart from standard Data Mining. Although the structure makes it more difficult to mine sequential data, there is also the reward to access information that can be contained in the structure of a sequence. Bautista-Thompson and Brito-Guevara [20] stress that the collective behaviour and the hidden relations between such data, can contain decisive information. Furthermore, they point out that the structure of sequences can have a certain dynamics (such as stationary, random, complex).

1) Tasks in Sequential Data Mining: Sequential Data Mining was primarily applied in the field of bioinformatics on genomic data and also, in the field of business intelligence on transactional data (especially from the retailer industry). Therefore, the following three tasks of Sequential Data Mining have emerged:

- Clustering: This task is about the grouping of unlabelled sequences into clusters. In general, this task is solved by combining well-known clustering algorithms with an appropriate distance function that is applied to sequences. Therefore the special properties of sequences and their structure need to be taken into account.
- Classification: This is most common task that includes building a classifier that is able to distinguish between two existing classes (labels). This is normally achieved by combining standard classification methods in conjunction with suitable feature construction techniques. The goal for the classification can be to decide if a sequence belongs to a certain class or if a sequence contains a subsequence of interest and its position (especially interesting for comparison of genomic data). The presented work focuses on the classification task.
- Hybrid: As the name suggests, this task is concerned with both: the identification of sequences classes as well as the characterization of the occurring sequential patterns [19].

2) Issues in Sequential Data Mining: Research in Sequential Data Mining usually revolves around so-called sequential databases in order to find sequential patterns. Therefore, sequential data mining research should consider the following four technical issues:

- Concept formulation: creating new concepts that lead to advances in the research field
- Design: creating novel techniques that are able to handle large volumes of data with a large number of dimensions. The techniques need to be able to handle the complex data structure while being able to take advantage of the underlying structure of the given sequential data.
- Optimizing cluster/classifying quality: modifying/altering existing techniques to achieve a better accuracy. Quality measurements in terms of classification are accuracy, precision and recall. In terms of clustering inter-precision or inter-cluster similarity are used.
- Optimizing pattern interestingness: this task aims to improve techniques in terms of their usefulness for the user. Measures include support, confidence, lift, novelty and actionability. Xing et al. [21] state that in addition to accurate quality results, the interpretability of sequence classifiers is both important and difficult

This research work deals with all four issues and focuses on the concept formulation as well as the design of a new sequence classification algorithm.

D. Problem description

As discussed in previous sections, the KDD process and Data Mining are about finding patterns in data. Initially these data comprised of static feature vectors that did not change over time [8]. The later years have brought more complex objects that need to be stored. The latest development in data collection and storage technologies allows companies to keep extremely large quantities of data relating to their daily activities [5]. This process introduced the temporal dimension into the field of Data Mining and allowed the storage of evolving (or dynamic) data over time. However, this dimension is neglected by most of the researchers: *'In Data Mining community, researchers pay little attention to time-stamps in temporal behavior [. . .] during classification'* [14]. This is quite a sub-optimal situation since *'knowledge about the behavior of objects is an integral part of understanding complex relationships in real-world systems and applications'* [8]. Time is necessary to markup complex behaviour. Kriegel argues that due to historical reasons (i.e., given their static data during the 1980s) many researchers created their algorithms only for static descriptions of objects and are therefore not designed to input data with dynamic behavior. The inclusion of dynamic properties of modern complex data models would allow revealing the information hidden in their temporal aspect and in addition to that, describe the relationship between complex objects. The type of information that is visible in the temporal dimension of a series of events is called sequential pattern.

Most of the sequence analysis work (see also Section II) is focused on finding frequent item sets, associate them with a certain order and then predict what items are bought next in a sequence. The research work described in this article is about finding a technique that is able to take the time span between a series of events into account and unveil hidden information that can be used for classification purposes.

This work will present an algorithm that is able to find discriminatory patterns in temporal based data and use them for classification purposes. The algorithms suggested so far have not been able to use the information hidden in sequential and time ordered data. Such information can be captured by creating sequence based features out of the original attributes of the given data set. The time dimension is thereby used in the construction step.

II. Related Work

Earlier work in the field of feature construction was done by Setiono and Liu [13]. They used a neuronal network to construct features in an automatic way for continuous and discrete data. Pagallo [15] proposed FRINGE, which builds a decision tree based on the primitive attributes to find suitable boolean combinations of attributes near the fringe of the tree. The newly constructed features are then added to the initial attributes set and the process is repeated until no new features are created. Zupan and Bohanec [16] used a neuronal net for attribute selection and applied the resulting feature set on the well known C4.5 [17] induction algorithm. Feature construction can also be used in conjunction with linguistic fuzzy rule models. García [18] et al. use previously defined functions over the input variables in order to test if

the resulting combination returns more information about the classification than the single variables. This process can lead to fuzzy rules of the following schema, which can include functions in the antecedent:

IF x_1 IS A_1AND $SUM(x_1, x_2)$ IS A_3 THEN Y IS B

A_3 and B represent fuzzy subset values that belong to the function's domain. The used function $SUM(x_1, x_2)$ is thereby treated as a new variable. However, in order to deal with increasing complexity of their genetic algorithm in the empirical part, García only used three functions ($SUM(x_i, x_j)$, $PRODUCT(x_i, x_j)$, $SUBSTRACT\_ABS(x_i, x_j)$) to enlarge the feature space. Another approach to feature construction, which utilizes a genetic algorithm, is described by Alfred [23]. Although, his approach is not using different functions to create new combinations of features, it can create a big variety of features since it is not limited to binary combination. That means that it is able to combine more than two attributes at a time. The genetic algorithm selects thereby the crossover points for the feature sequences. Sia [24] proposes a 'Fixed-Length Feature Construction with Substitution' method called FLFCWS. It constructs a set that consist of randomly combined feature subsets. This allows initial features to be used more than once for feature construction.

The next two subsections present the two most famous Feature Construction techniques for sequential data in greater detail.

A. MFE3/GA

Shafti [7] presents MFE3/GA (Multi-Feature Extraction using GA), a method that uses a global search strategy (i.e., finding the optimal solution) to reduce the original data dimensions and find new non-algebraic representations of features. Her primary focus is to find interactions between the original attributes (such as the interaction of several cards in a poker game that form a certain hand). MFE3/GA basically searches through the initial space of attribute subsets to find subset of interaction attributes as well as a function over each of the found subsets. The suitable functions are then added as new features to the original data set. The C4.5 learner is then applied for the data mining process. So far only nominal attributes are being processed, so that class labels and binary/continuous attributes need to be normalized. A feature is thereby a bit-string of length N, where each bit shows the presence or absence of one of the N original attributes. Subsets of these features are associated with a function defined over the attributes in the subset. This allows a non-algebraic (operator-free) representation of the original attributes. The output of the associated functions f_i for each subset $S_i = (x_{i_1}, \ldots, x_{i_m})$ are basically the binary class labels. The labels are retrieved from the training samples that match the subset. It can be possible that both labels for one subset are occuring in the training data. In this case a so called mixed-tuple label can be associated with the subset (other labels are types pure and unknown).

B. FeatureMine

Lesh, Zaki and Ogihara present FeatureMine [12] - another well known feature construction technique for sequential data. It combines two data mining paradigms: sequence mining and classification algorithms. They understand sequences as a series of events, e.g., $AB \rightarrow B \rightarrow CD$. There is also a timestamp associated with each event. FeatureMine starts by mining frequent and strong patterns within the sequences. Frequency is defined by a threshold that is specified by the user. Strong is defined as a confidence level that needs to be over a user specific threshold. The found frequent sequence patterns are pruned and selected using some heuristics. The prevailing sequences lattices are stored in a matrix $n * m$ database layout, whereby the rows n represents the sequences and the columns m represent the prevailing sequence lattices. The cells of the matrix contain and boolean indicator if a sequence contains the corresponding sequence lattice. The constructed features are associated with a class label and then feed into the Naive Bayes classification algorithms.

C. Contribution

We propose an automated algorithm that is able to systematically construct and assess suitable new features based on data sequences for binary classification tasks. It thereby is also able to utilize the time dimension in a sequence of events in order to access information, which can have a significant impact on the discriminatory power of features. Thereby, the algorithm transforms sequential data into tuple-based data in a way, that allows standard algorithm such as Neuronal Networks, Bayesian Belief Network, Decision Trees or Naive Bayes to be applied on sequential data.

So far, feature construction techniques build new features by combining columns of a data set (i.e., 'horizontally'). We also apply these techniques with a larger variety of mathematical operators. In addition to that, we are able to utilize the time elapsed between data points. Our approach is novel, since we include the vertical dimension of data, i.e., the rows of a sequence, in order to create new features. This is achieved by combining numeric values (or its probabilities in terms of string attributes) of the corresponding occurrences. The original values are aggregated during the feature construction process. This allows to store sequence based information on tuple level. As a result of that, the above mentioned standard algorithms can be applied (not all are able to handle sequenced data sets right away).

The proposed techniques are extending the given problem-space and search for a combination of dimensions that allow to separate the binary classes that need to be classified. It thereby utilizes abstracted patterns that can occur in the data and is able to validate the created combinations.

III. General characteristics of sequential data

This work often refers to the term sequential data. Thereby, we understand data, that can be ordered by time and can be grouped to logical units (i.e., the sequence). A simple example for that are sessions in an online shop. Customers

TABLE I: Schema of sequence data

r	t	s_{id}	a_1	a_2	...	a_i	s_{label}
r_1	t_1	s_{id_1}	a_{1_1}	a_{2_1}	...	a_{i_1}	0
r_2	t_2	s_{id_1}	a_{1_2}	a_{2_2}	...	a_{i_2}	0
r_3	t_3	s_{id_1}	a_{1_3}	a_{2_3}	...	a_{i_3}	0
r_4	t_4	s_{id_2}	a_{1_4}	a_{2_4}	...	a_{i_4}	1
r_5	t_5	s_{id_2}	a_{1_5}	a_{2_5}	...	a_{i_5}	1
r_6	t_6	s_{id_2}	a_{1_6}	a_{2_6}	...	a_{i_6}	1
...	...	...	...	...	...	...	...
r_m	t_m	s_{id_n}	a_{1_m}	a_{2_m}	...	a_{i_m}	...

can view products and put them into their shopping basket. Every action can be represented in a data set E as a row r with several attributes $a_i \in E$. Each row is provided with a timestamp t. A row can be associated to a logical unit s_{id} (in our case the session id). There are n sequences s_{id_n} in a data set E. Each sequence s_{id_n} consist of at least one row r. The number of rows in a sequence equals to the length of a sequence ls, so that $1 \leq ls \leq m$. Table I depicts the general schema of sequential data: It is important to differentiate between the number of rows (or tuples) m of a data set and the number of sequences n. Sequence s_{id_1}, for example, has a length ls of three and contains a matrix such as $s_{id_1} = \begin{pmatrix} a_{1_1} & a_{2_1} & \dots & a_{i_1} \\ a_{1_2} & a_{2_2} & \dots & a_{i_2} \\ a_{1_3} & a_{2_3} & \dots & a_{i_3} \end{pmatrix}$

In order to use our proposed method, which is described in detail in the following section, the user has also to annotate the following columns on a data set:

- t: timestamp column that is used for temporal based features. It is used to calculate the time elapsed between the collected data points of a sequence.
- s_{id}: sequence identifier column that is used for sequence aggregation. It identifies events/objects that can be logically associated to one entity
- s_{label}: the proposed algorithm requires a binary column as target value. This is needed in order to automatically calculate the information gain of newly constructed features. Every sequence must only have one label, i.e., a customer in an online shop is either a returning customer or not (it can not be both at the same time).

During the feature construction process, we will create a feature table, which includes the s_{id}, s_{label} and the created features $f_p \in S$. S is thereby defined as a set of constructed features. Please refer to Table II, for a schema of such a table.

TABLE II: Schema of feature table

s_{id}	f_1	f_2	...	f_p	s_{label}
s_{id_1}	f_{1_1}	f_{2_1}	...	f_{p_1}	0
s_{id_2}	f_{1_2}	f_{2_2}	...	f_{p_2}	1
...	...	...	...	...	...
s_{id_n}	f_{1_n}	f_{2_n}	...	f_{p_n}	...

The data sequences are aggregated on a tuple-based level. This enables the application of many standard classification algorithms.

IV. Feature Construction for Data Sequences

Our goal is to extend and search the initial problem space as much as possible. Problem space is thereby defined through the primitive (original) attributes E, which are used to solve a binary classification task. The accessible feature space expands, if more features are constructed. Albeit, this leads to an increase in search time, it brings a higher chance to find discriminatory features. In order to keep things as simple as possible, we describe the algorithm in five different subsections, each describing a certain sort of features construction technique. Please note that the initial attributes are, in a first step, categorized in string and numeric attributes. Reason for this is, that not all described functions are applicable on string values. Note, that after each feature construction technique, we normalize the newly generated features with min-max normalization, depicted in (1). This provides an easy way to compare values that are on different numerical scales or different units of measure.

$$Normalized(e_i) = \frac{e_i - E_{min}}{E_{max} - E_{min}}, for\ E_{max} > E_{min} \quad (1)$$

The first Subsection IV-A will show construction techniques for both string and numeric attributes. The second Subsection IV-B describes construction techniques for string-only attributes. After that we will focus in the third Subsection IV-C on numeric-only construction techniques. Subsection IV-D describes temporal based feature construction techniques. This section is concluded by Subsection IV-E, which describes feature construction based on sequence distribution.

A. Distinct occurrences based features

The general idea for this feature construction technique is to analyze if different occurrences per sequence allows to discriminate between the given labels. Basically, we aggregate all sequences s_{id_n} and count the distinct occurrences (so no duplicates are counted) for each given string as well as for each numeric attribute a_{i_m}. The constructed features f_p are then collected in S, together with its corresponding sequence identifier s_{id} and the corresponding session label s_{label}. Please note that the sequence identifier s_{id} is unique in S (as opposed to E). The corresponding pseudo-code is depicted in Fig. 1.

In order to assess the quality of the new constructed feature f_i, we calculate two measurements in order to assess the quality. The first one is the average of all aggregated values per label $s_{label} \in \{0, 1\}$. The normalized difference between both averages is called split and is calculated as depicted in (4).

$$avg_0 = \quad avg(\{f_p \in S | s_{label} = 0\}) \quad (2)$$

$$avg_1 = \quad avg(\{f_p \in S | s_{label} = 1\}) \quad (3)$$

$$split_{f_i} = \quad \frac{|avg_0 - avg_1|}{avg_0 + avg_1} \quad (4)$$

The second measurement to assess the quality of the constructed features is the number of zero and NULL values for each target label. This is a support measurement that denotes if the achieved split value is based on many sequences or not.

Input: E // set of nominal and continuous attributes
$s_{label} \in \{0, 1\}$ // binary label indication
Def: $a_i \in E$ // single attribute or a column in a data set
$s_{id} = (r_1, r_2, \ldots, r_m)$ // sequences of rows r_i
$S = \emptyset$ // set of constructed features
for each $a_i \in E$ {
for each $s_{id} \in E$ {
$f_p := (|\{a_{i_n}\}|, s_{id}, s_{label})$
$S := S \cup f_p$
}
}
return S

Fig. 1: Pseudo-code feature construction based on distinct occurences per label

So there could be the situation that a constructed feature has a high split value, but might be useless since it cannot be used very often due to large number of 0 values for the particular features.

B. Concatenation based features

The purpose of this type of feature construction is to highlight simpler interactions among data. We systematically concatenate every string attribute in pairs of two and then again, count the distinct value-pairs per sequence identifier. Thereby, interactions such as, if a_1 AND a_2 have low value-pair variety for label 0, but a high value-pair variety for label 1, are highlighted. Even for data sets with a high number of different occurrences, this kind of feature construction will highlight distinct occurrences between both labels. This procedure is only applicable on string attributes. This approach is similar to most common column combinations that is described widely in the literature (e.g., [7], [16], [23]). However, we once again use this technique on a different abstraction layer since we aggregate via the sequence identifier s_{id}. The corresponding pseudo-code is depicted in Fig. 2.

The algorithm copies the input attribute list E for looping purposes into a second variable E_2. Right after the second loop, it deletes the current attribute from the copied list ($E_2 - a_{2i}$). Reason for this is to avoid the same features to occur twice, due to symmetric properties. If, for example, we combine column $a_i = X$ and $a_j = Y$ of a data set, we will yield feature XY. This feature will have the same variability per sequence as the vice versa feature YX. The construction of such features can be avoided by deleting the current feature from the copied feature list E_2.

C. Numeric operator based features

The basic idea of this feature construction technique is to combine two numeric attributes with basic arithmetic operators such as '+', '-', '*' or '/'. Garcia [18] and Pagallo [15] for instance are using similar techniques with fewer operators. In addition to the repeated use of arithmetic operators we, once again, use the sequence identifier attribute to aggregate the constructed features for each sequence. Lets put this into an

Input: E // set of nominal attributes
$s_{label} \in \{0, 1\}$ // binary label indication
Def: $a_i \in E$ // single attribute or a column in a data set
$s_{id} = (r_1, r_2, \ldots, r_m)$ // sequences of rows r
$S = \emptyset$ // set of constructed features
$E_2 = E$ // copy of E, used for looping
$con()$ // concatenates two values
for each $a_i \in E$ {
//remove a_i to avoid vice versa features
$E_2 := E_2 - \{a_i\}$
for each $a_j \in E_2$ {
for each $s_{id} \in E$ {
$f_p = (|(con(a_i, a_j))|, s_{id}, s_{label})$
$S = S \cup f_p$
}
}
}
return S

Fig. 2: Pseudo-code feature construction based on concatenated string attributes

example: attributes a_i and a_j are combined with the multiplication operator '*' for a sequence s_{id_1}. The resulting feature $f = a_i * a_j$ is derived from the sequence $s_{id_1} = \begin{pmatrix} a_{i_1} & a_{j_1} \\ a_{i_2} & a_{j_2} \\ a_{i_3} & a_{j_3} \end{pmatrix}$

The sequence consists of three data points. In the aggregation phase, we sum up the multiplied attributes for all sequences $\sum_{j=1}^{3} f_{ij}$. This process is repeated for all possible combinations of numeric attributes for all of the above mentioned mathematical operators. The full pseudo-code is depicted in Fig. 3. For this technique, we also avoid vice versa features as described in previous Subsection IV-B.

D. Temporal axis based features

The general idea for this feature construction technique is to use the time axis, which is given in each sequence by the time indicator column t. This is applicable for both, numeric as well as string attributes. However, for string attributes, there needs to be some preparations done, which are explained further down in this subsection. We continue here to describe the process for numeric attributes. What the algorithm basically does, is to multiply the time interval (e.g., days, hours, minutes), between earliest data point and the current data point with the numeric value of corresponding attribute, which results in a weighting.

Table III shows this for two example sequences. We have two attributes a_i and a_j for two sequences as well as the t column. In order to calculate the temporal based feature for attribute sequence $s_{id} = 1$ in terms of attribute a_i, we first have to calculate the time between the earliest data point $min(t)$ with $t \in sequence(s_{id})$ and each of the 'current' data points t. In Table III, this is depicted by the $\Delta time\_in\_days$ column. The next step is to multiply the value of each t_i in $s_{id} = 1$ with

Input: E // set of primitive numeric attributes
$s_{label} \in \{0, 1\}$ // single value label indication
Def: $a_i \in E$ // single attribute or a column in a data set
$s_{id} = (r_1, r_2, \ldots, r_m)$ // sequences of rows r
$S = \emptyset$ // set of constructed features
$E_2 = E$ // copy of E, used for looping
O // set of arithmetic operators
ls // length of a sequence s_{id}
for each $a_i \in E$ {
//remove a_i to avoid vice versa features
$E_2 := E_2 - \{a_i\}$
for each $a_j \in E_2$ {
for each $o \in O$ {
for each $s_{id} \in E$ {
$f_p = (\sum_{i=1}^{ls}(a_i \ o \ a_j), s_{id}, s_{label})$
$S = S \cup f_p$
}
}
}
}
return S

Fig. 3: Pseudo-code feature construction based on numeric attributes

its corresponding delta time value: $(a_{i_1} * 1, a_{i_2} * 11, \ldots, a_{i_4} * 24)$. The sum of this value is the new time based constructed feature f_p. This process is repeated for all sequences s and for all numerical attributes E.

TABLE III: Example for creating temporal based features

s_{id}	t	$min(t)$ per s_{id}	$\Delta time - in\ days$	a_i	a_j	s_{label}
1	01.01.2013	01.01.2013	1	a_{i_1}	a_{j_1}	0
1	10.01.2013	01.01.2013	11	a_{i_2}	a_{j_2}	0
1	15.01.2013	01.01.2013	16	a_{i_3}	a_{j_3}	0
1	23.01.2013	01.01.2013	24	a_{i_4}	a_{j_4}	0
2	24.01.2013	24.01.2013	1	a_{i_5}	a_{j_5}	1
2	28.01.2013	24.01.2013	5	a_{i_6}	a_{j_6}	1
2	30.01.2013	24.01.2013	7	a_{i_7}	a_{j_7}	1

However, there are two directions of including the time for this feature construction technique. What we described above puts a stronger emphasis on the recent history. It is also possible to increase the weight of the past by using the (max_date - current_date) operator to calculate the $\Delta time\_in\_days$ column. An example of this is depicted in Table IV. The complete pseudo code is depicted in Fig. 4.

The above mentioned techniques are applicable on numeric attributes. For string attributes, it is possible to replace the string by the posterior probability $p(\theta|x)$ (see also Hand [26], pp. 117-118 and pp. 354-356). Thereby, θ represents the probability of the parameters for a given evidence x. In our example case, we have the distribution of our two labels as parameters θ and occurrences of a_i as evidence x.

Based on this the posterior probability can be calculated as

TABLE IV: Example for creating temporal based attributes with a stronger emphasis on the distant past

s_{id}	t	$max(t)$ per s_{id}	$\Delta time - in\ days$	a_i	a_j	s_{label}
1	01.01.2013	23.01.2013	24	a_{i_1}	a_{j_1}	0
1	10.01.2013	23.01.2013	14	a_{i_2}	a_{j_2}	0
1	15.01.2013	23.01.2013	9	a_{i_3}	a_{j_3}	0
1	23.01.2013	23.01.2013	1	a_{i_4}	a_{j_4}	0
2	24.01.2013	30.01.2013	7	a_{i_5}	a_{j_5}	1
2	28.01.2013	30.01.2013	3	a_{i_6}	a_{j_6}	1
2	30.01.2013	30.01.2013	1	a_{i_7}	a_{j_7}	1

Input: E // set of continuous/numeric attributes
t // time indicator column
$s_{label} \in \{0, 1\}$ //binary label indication
Def: $a_i \in E$ // single attribute or a column in a data set
$s_{id} = (r_1, r_2, \ldots, r_m)$ // sequences of rows r
$S = \emptyset$ // set of constructed features
$E_2 = E$ // copy of E, used for looping
ls // length of a sequence s_{id}
$max()$ // returns max value of a set
for each $a_i \in E$ {
for each s_{id} {
$f_p = (\sum_{i=1}^{ls}((\max_{k=1,\ldots,ls}(t_k) - t_i) * a_i), s_{id}, s_{label})$
$S = \{S \cup f_p\}$
}
}
return S

Fig. 4: Pseudo-code feature construction of temporal based attributes

depicted in (5)

$$p(s_{label} = 1|a_i) = \frac{p(a_i|s_{label}=1) * p(s_{label}=1))}{p(a_i)} \quad (5)$$

In order to apply this for string based attributes, we can construct new features f for string attributes as depicted in (6)

$$f_p = \sum_{i=1}^{ls}(\max_{k=1,\ldots,m}(t_k) - t_i) * (p(s_{label} = 1|a_i)) \quad (6)$$

If there are occurrences in the data that have a great tendency towards a particular label (i.e., having a high probability for one label), we can make this pattern visible by multiplying the posterior possibility with the temporal axis of the given sequence.

However, if there are too many different occurrences, lets say more than 1.000 different values per attribute, this technique could have problems dealing with very small probabilities. So, it is recommended to take the logarithm of the posterior probability for cases with high cardinality.

E. Sequence distribution based features

It is also possible that a discriminatory pattern evolves around distributions of numeric values in the given sequences.

Input: E // set of continuous/numeric attributes
$s_{label} \in \{0, 1\}$ // binary label indication
Def: $a_i \in E$ // single numeric attribute in a data set
$s_{id} = (r_1, r_2, \ldots, r_m)$ // sequences of rows r
$S = \emptyset$ // set of constructed features
$E_2 = E$ // copy of E, used for looping
O // set of arithmetic operators
for each $a_i \in E$ {
for each $s_{id} \in E$ {
$f_p = (STD\_DEV(s_{id}), s_{id}, s_{label})$
$f_p = f_p \cup (VAR(s_{id}), s_{id}, s_{label})$
$f_p = f_p \cup (AVG(s_{id}), s_{id}, s_{label})$
$S = S \cup f_p$
}
}
return S

Fig. 5: Pseudo-code feature construction based on sequence distribution

Therefore, this feature construction technique is focusing on patterns that are based on variability, standard deviation and average. This construction techniques highlights patterns as for example:

- one numeric value of a class is oscillating while the value is stable for the other class
- the values for one class are more spread out than for the other class
- the average value of an attribute per sequence of a certain class is in general higher or lower, then of the other class

In principle, we calculate the above mentioned values for each sequence of each numeric attribute in a data set. The full pseudo-code is depicted in Fig. 5.

V. Experimental setup and Results

This section is divided into three subsection in which we will first look at the technical framework we used during our experiments. This is followed by a brief look at the data profile and the corresponding classification task. The third subsection will then compare and discuss the results of our experiments.

A. Technical Framework and Infrastructure

All implementations and experiments were carried out on a Microsoft Windows Server 2008 R2 Enterprise Edition (6.1.7601 Service Pack 1 Build 7601) with four Intel Xeon CPUs E5320 (1.86 GHz, 1862 MHz). The available RAM comprised of 20 GB installed physical memory and 62 GB virtual memory (size of page file 42 GB). The widespread freeware data mining software RapidMiner (version 5.2.008) was used for the standard methods under comparison: Decision Tree, Naive Bayes, Neuronal Network and Random Forrest (for a closer description please also see Witten [25] pp. 191-294, Han [3] pp. 291-337). The method Bayesian Belief Network required the installation of the free RapidMiner extension WEKA. We used the default parameters for all of the above mentioned classification algorithms.

B. Data Profile

The data we used for our experiments was retrieved from the DataMiningCup 2013. The training as well as the test data set can be downloaded on the following site: http://www.data-mining-cup.de/en/review/dmc-2013/. The given historical data from an online shop consisting of session activities from customers. The goal of the task is to classify sessions into a buyer and a non-buyer class. The parameters of the train data was predefined by the task of the DataMiningCup 2013 and are as follows:

- total number of rows: 429,013
- number of sessions: 50,000
- number of numeric attributes: 21
- number of string attributes: 2

The test data was also given by the DataMiningCup requirements, which had the following parameters:

- total number of rows: 45,068
- number of sessions: 5,111
- number of numeric attributes: 21
- number of string attributes: 2

Most of the given attributes are numeric. Please note that there is no exact time column given. Therefore, we used a artificial id column to map the temporal order of the various sessions. We also used this column to calculate the temporal based features described in Subsection IV-D.

C. Comparison of original attributes vs constructed features sets

As a first step, we used the given primitive attributes to solve the task. We used the accuracy measurement (7) due to a similar label distribution (45 % to 55 %) and both labels are associated with the same 'costs' for misclassification.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

As it can be seen in Fig. 6, the Naive Bayes classification algorithm was able to achieve better result than the base line (the other algorithms defaulted and predicted label = 0 for all sessions). The Bayesian Belief Networks are not applicable for situations in which the same s_{id} can occur several time (therefore a accuracy rate of 0 %). In a next step, we used our suggested feature construction algorithm in order to aggregate the sessions and find useful features. During this process, a grand total of 860 features were created:

- # of distinct occurrences based features: 19
- # of string concatenation based features: 2
- # of arithmetic based features: 760
- # of temporal axis based features: 20
- # of sequence distribution based features: 59

All constructed features were normalized with the min-max normalization. They were, in a first series of experiments, assessed by calculating the split value for each feature. The

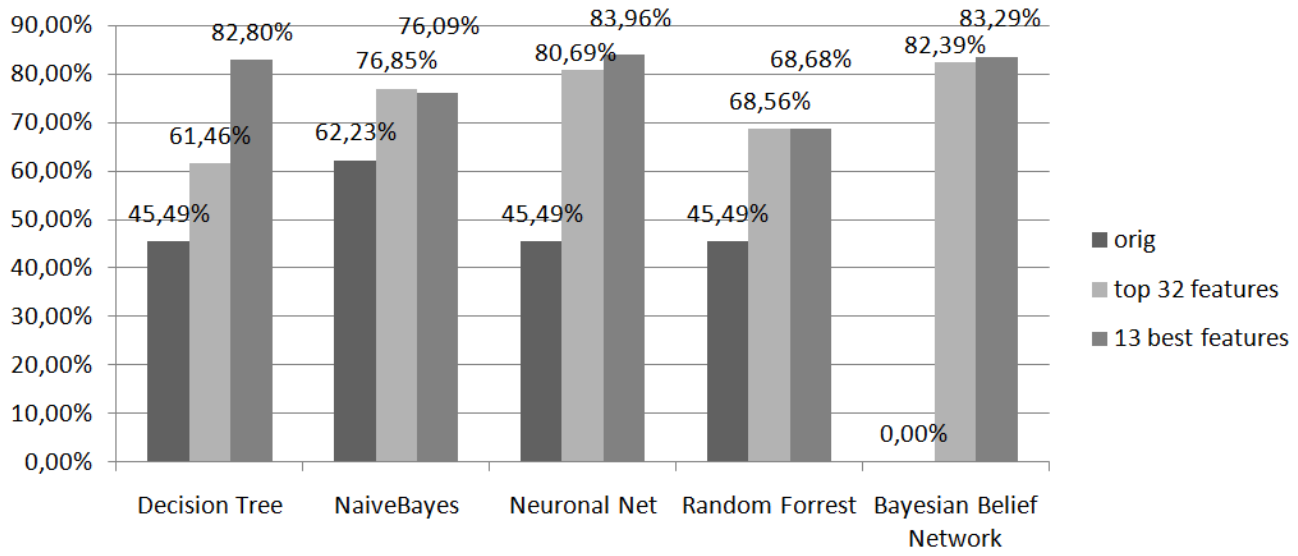


Fig. 6: Accuracy rate comparison original data set with primitive attributes variations of constructed features.

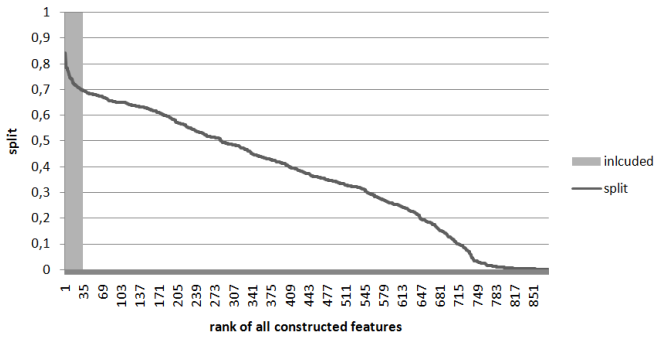


Fig. 7: All constructed features ranked by their split value.

features were ranked by their split value, as it can be seen in Fig. 7. The best feature achieve a split value of 0.843, the lowest of 0.0003. In order to keep execution times low, we chose only the top 32 constructed features from the ranked list for our second run. Fig. 6 shows the impressive improvement for the compared standard methods. Since the s_{id} is unique for the constructed features set, the Bayesian Belief Networks are applicable.

However, focusing only on the split measurement for feature selection is not enough. In a second range of experiments, we only included those features, which achieved a minimum split value of 0,70 and a had a minimum support value of 0,50. A total number of 13 features met these criteria (10 operator numeric and 3 sequence distribution based features). The results for the best feature are shown in Fig. 6. It can be seen that the smaller constructed feature space is able to perform better or at least as good as the top 29 features only ranked by split. This shows that complex problems with sequential data can be simplified and solved by features construction. We can also see that for this data set, operator numeric features turned out to be the most benefiting ones. Reason for this is that there were only two string attributes in the original data set as well as the lack of a proper timeline (see also Section V-B). This means that in this data set, the pattern to distinguish between the two given labels is not that dependent on the temporal dimension than in other data sets (e.g., [1]). This also highlights the importance of the presented features selection techniques. Without them, arbitrary and useless features would have mislead the used classifiers.

VI. Conclusion

Data pre-processing and selection are important steps in the data mining process. This can be challenging, if there is no domain expert knowledge available. The algorithm proposed in this work helps, not only to understand the patterns within the data, but also, to simplify more complex data structures (such as sequential data). This is achieved by various aggregation and combination techniques that allow to increase the feature space of a given data set and eventually, to highlight present feature interactions. The feature construction algorithm can be applied in conjunction with well known standard algorithms and boosts classification performance in a big variety of fields with similar specifications (such as the detection of credit card fraud, network intrusions, bots in computer games). Its systematic approach can also help domain experts to find previously unknown interactions among data and therefore, to get a better understanding of their domain.

VII. Future Work

Further ways for extending the features space could be to implement more numerical features generated by logarithm, exponential function or combinations of more than two attributes. The algorithm itself could be optimized to assess the quality of a candidate feature before actually calculating

it. Another development direction could be to align the constructed features in a way, that would allow to classify data without the help of one of the standard algorithms.

REFERENCES

[1] M. Schaidnagel and F. Laux, "Feature construction for time ordered data sequences," in *Proceedings of the Sixth International Conference on Advances in Databases, Knowledge, and Data Applications*, Chamonix, April 20-24, 2014, pp. 1-6.

[2] W. Lee, "A Data mining framework for constructing features and models for intrusion detection systems," PhD thesis, Columbia University, Graduate School of Arts and Sciences, 1999.

[3] J. Han and M. Kamber, *Data mining: concepts and techniques* 2. edition pp. 48-97 second edition, San Francisco, Morgan Kaufmann, 2006.

[4] H. Liu and H. Motoda, *Feature extraction, construction and selection: a data mining perspective*, Boston, Kluwe Academic Publisher, 1998.

[5] W. Lin, M. Orgun, and W.J. Graham, "An overview of temporal data mining," in *Proceedings of the 1st Australian data mining workshop*, Canberra, Australia, 2002, pp. 83-90.

[6] L. S. Shafti and E. Pérez, "Constructive induction and genetic algorithms for learning concepts with complex interaction," in *Proceedings of The Genetic and Evolutionary Computation Conference*, Washington, June 2005, pp. 1811-1818.

[7] L. S. Shafti and E. Pérez, "Data reduction by genetic algorithms and non-algebraic feature construction: a case study," in *Proceedings of: Eighth International Conference on Hybrid Intelligent Systems*, Barcelona, September 2008, pp. 573-578.

[8] H.P. Kriegel, K. M. Borgwardt, P. Krger, A. Pryakhin, M. Schubert, and A. Zimek, "Future trends in data mining," in *Data Mining and Knowledge Discovery*, vol. 15, no. 1, Springer, 2007, pp. 87-97.

[9] I. Guyon, S. Gunn, M. Nikravesh, and L. A. Zadeh, *Feature extraction: foundations and applications*, Berlin, Springer, 2006.

[10] K. Cios, W. Pedrycz, R. W. Swiniarski, and L. A. Kurgan,*Data mining: a knowledge discovery approach*, New York, Springer US, 2007.

[11] S. Markovitch and D. Rosenstein, "Feature generation using general constructor functions," in *Machine Learning*, vol. 49, no. 1, Kluwer Academic Publishers, 2002, pp. 59-98.

[12] N. Lesh, M. J. Zaki, and M. Ogihara, "Scalable feature mining for sequential data," in *IEEE Intell. Syst.* No. 2, 2000, pp. 48-56.

[13] R. Setiono and H. Liu, "Fragmentation problem and automated feature construction," in *Proceedings of: 4th Conference on Data Mining and Optimization (DMO)*, Langkawi, September 2012, pp. 53-58.

[14] Y. Yang, L. Cao, and L. Liu, "Time-sensitive feature mining for temporal sequence classification," in *Proceedings 11th Pacific Rim International Conference, Wellington*, December 2013, pp. 315326.

[15] G. Pagallo, "Learning DNF by decision trees," *Machine Learning*, pp. 71-99 Kluwer Academic Publishers, 1990.

[16] B. Zupan and M. Bohanec, "Feature transformation by function decomposition," in *Journal IEEE Intelligent Systems archive*. Volume 13 Issue 2, March 1998, pp. 38-43.

[17] J.R. Quinlan, "C4.5: programs for machine learning". Morgan Kaufmann, 1993.

[18] D. García, A. González, and R. Pérez, "A two-step approach of feature construction for a genetic learning algorithm," in *Proceedings of: IEEE International Conference on Fuzzy Systems*, Taipei, June 2011, pp. 1255-1262.

[19] G. Dong and J. Pei, *Sequence data mining*, New York, Springer US, 2007.

[20] E. Bautista-Thompson and R. Brito-Guevara, "Classification of data sequences by similarity analysis of recurrence plot patterns," in *Proceedings of Seventh Mexican International Conference on Artificial Intelligence*, Tuxtla Gutirrez, Mexico, 2008.

[21] Z. Xing, J. Pei, and E. Keogh, "A brief survey on sequence classification," in *ACM SIGKDD Explorations Newsletter*, No 1, 2010, pp. 40-48.

[22] D. García, Antonio González, and R. Pérez, "An iterative strategy for feature construction on a fuzzy rule-based learning algorithm," in *Proceedings of: 11th International Conference on Intelligent Systems Design and Applications*, Cordoba, November 2011, pp. 1235-1240.

[23] R. Alfred, "DARA: data summarisation with feature construction," in *Proceedings of: Second Asia International Conference on Modelling & Simulation*, Kuala Lumpur, May 2008, pp. 830-835.

[24] F. Sia and R. Alfred, "Evolutionary-based feature construction with substitution for data summarization using DARA," in *Proceedings of: fourth Conference on Data Mining and Optimization (DMO)*, Langkawi, September 2012, pp. 53-58.

[25] I. Witten and F. Eibe, *Data mining : practical machine learning tools and techniques* 2. edition, San Francisco, Morgan Kaufmann, 2005, pp. 48-97.

[26] D. Hand, H. Mannila, and P. Smyth, *Principles of Data Mining*, MIT Press, 2001.

Virtual-BFQ: A Coordinated Scheduler to Minimize Storage Latency and Improve Application Responsiveness in Virtualized Systems

Alexander Spyridakis, Daniel Raho, and Jérémy Fanguède
Virtual Open Systems
Grenoble - France
Email: {a.spyridakis, s.raho, j.fanguede}@virtualopensystems.com

***Abstract*—Preserving responsiveness is an enabling condition for running interactive applications effectively in virtual machines. For this condition to be met, low latency usually needs to be guaranteed to storage-Input/Output operations. In contrast, in this paper we show that in virtualized environments, there is a missing link exactly in the chain of actions performed to guarantee low storage-I/O latency. After describing this problem theoretically, we show its practical consequences through a large set of experiments with real world-applications. After detailing the possible solutions to replace this missing connection, we detail our chosen solution based on the I/O scheduler BFQ (Budget Fair Queueing) named Virtual-BFQ. Which it designed to preserve a high application responsiveness in KVM (Kernel-based Virtual Machine) virtual machines on ARM architectures. For the experiments, we used two Linux schedulers, both designed to guarantee a low latency, and a publicly available I/O benchmark suite, extended to be used also in a virtualized environment. As for the experimental testbed, we ran our experiments on the following three devices connected to an ARM embedded system: an ultra-portable rotational disk, a microSDHC (Secure Digital High Capacity) Card and an eMMC (embedded Multimedia card) device. This is an ideal testbed for highlighting latency issues, as it can execute applications with about the same I/O demand as a general-purpose system, but for power-consumption and mobility issues. According to the experimental results reported in this paper, even in the presence of a heavy background workload on the guest virtual disk, plus a heavy additional background workload on the physical storage device corresponding to that virtual disk, Virtual-BFQ does preserve in the guest a high application responsiveness.**

Keywords–KVM/ARM; virtualization; responsiveness and soft-real time guarantees; coordinated scheduling; embedded systems; Virtual-BFQ.

I. INTRODUCTION

Virtualization is an increasingly successful solution to achieve both flexibility and efficiency in general-purpose and embedded systems. However, for virtualization to be effective also with interactive applications, the latter must be guaranteed a high, or at least acceptable responsiveness. In other words, it is necessary to guarantee that these applications take a reasonably short time to start, and that the tasks requested by these applications, such as, e.g., opening a file, are completed in a reasonable time.

To guarantee responsiveness to an application, it is necessary to guarantee that both the code of the application and the I/O requests issued by the applications get executed with a low latency. In virtualized systems the responsiveness is not always guarantees [1], and expectedly, there is interest and active research in preserving a low latency in virtualized environments [2][3][4][5][6][7][8], especially in soft and hard real-time contexts [9][10][11]. In particular, some virtualization solutions provide more or less sophisticated Quality of Service mechanisms also for storage I/O [2][3][12][13]. However, even just a thorough investigation on application responsiveness, as related to storage-I/O latency, seems to be missing. In this paper, we address this issue by providing the following contributions.

A. Contributions of this paper

First, we show, through a concrete example, that in a virtualized environment there is apparently a missing link in the chain of actions performed to guarantee a sufficiently low I/O latency when an application is to be loaded, or, in general, when any interactive task is to be performed. To this purpose, we use as a reference two effective schedulers in guaranteeing a high responsiveness: Budget Fair Queuing [14] and Completely Fair Queuing [15]. They are two production-quality storage-I/O schedulers for Linux.

Then, we report experimental results with real-world applications. These results confirm that, if some applications are competing for the storage device in a host, then the applications running in a virtual machine executed in the same host may become from not much responsive to completely unresponsive. To carry out these experiments, we extended a publicly available I/O benchmark suite for Linux [16], to let it comply also with a virtualized environment.

The solution described in this paper solves the problem highlighted previously, it is an extension of the BFQ storage I/O scheduler [17]. Such an extension can be implemented in several ways. So first, we provide an analysis of the solution space. From this analysis, we highlight the solution that seems to provide most benefits. We did implement such a solution and named Virtual-BFQ (V-BFQ) [18] the resulting extended version of BFQ. We describe its implementation in detail. And we report our experimental results with this scheduler. The results obtained confirm that V-BFQ does preserve a high application responsiveness in a virtualized environment even with the presence of heavy background workloads.

As an experimental testbed, we opted for an ARM embedded system, based on the following considerations. On one

hand, modern embedded systems and consumer-electronics devices can execute applications with about the same I/O demand as general-purpose systems. On the other hand, for mobility and energy-consumption issues, the preferred storage devices in the former systems are (ultra) portable and low-power ones. These devices are necessarily slower than their typical counterparts for general-purpose systems. Being the amount of I/O the same, the lower the speed of a storage device is, the more I/O-latency issues are amplified. Finally, as a virtualization solution we used the pair QEMU (Quick EMUlator) and KVM, one of the most popular and efficient solutions in ARM embedded systems.

B. Organization of this paper

In Section II, we describe the schedulers that we use as a reference in this paper. Then, in Section III we show the important I/O-latency problem on which this paper is focused. After that, in Section IV, we describe how we modified the benchmark suite to execute our experiments. And in Section V, we report our experimental results that highlight the latency problem. Then, in Section VI we provide an analysis of the possible solution space, and highlight the solution that apparently provides the best trade-off between pros and cons. After that, we describe V-BFQ in detail in Section VII. And then in Section VIII, we report our results with V-BFQ for the same experiments that we executed for BFQ and CFQ. Finally, in Section IX we compared the results obtained with V-BFQ with two other standard I/O schedulers for Linux : *Deadline* and *NOOP*.

II. Reference schedulers

To show the application-responsiveness problem that is the focus of this paper, we use the following two storage-I/O schedulers as a reference: BFQ [17] and CFQ [15]. We opted for these two schedulers because, they, both guarantee a high throughput and low latency. In particular, BFQ achieves even up to 30% higher throughput than CFQ on hard disks with parallel workloads. Strictly speaking, only the second feature is related to the focus of this paper, but the first feature is however important, because a scheduler achieving only a small fraction of the maximum possible throughput may be, in general, of little practical interest, even if it guarantees a high responsiveness. The second reason why we opted for these schedulers is that up-to-date and production-quality Linux implementations are available for both. In particular, CFQ is the default Linux I/O scheduler, whereas BFQ is being maintained separately [16]. In addition to the extended tests for BFQ and CFQ, we also identified similar behaviour with the Noop and Deadline schedulers. In the next two sections, we briefly describe the main differences between the two schedulers, focusing especially on I/O latency and responsiveness. For brevity, when not otherwise specified, in the rest of this paper we use the generic term *disk* to refer to both a hard disk and a solid-state disk.

A. BFQ

BFQ achieves a high responsiveness basically by providing a high fraction of the disk throughput to an application that is being loaded, or whose tasks must be executed quickly. In this respect, BFQ benefits from the strong fairness guarantees it provides: BFQ distributes the disk throughput (and not just the disk time) as desired to disk-bound applications, with any workload, *independently of* the disk parameters and even if the disk throughput fluctuates. Thanks to this strong fairness property, BFQ does succeed in providing an application requiring a high responsiveness with the needed fraction of the disk throughput in any condition. The ultimate consequence of this fact is that, regardless of the disk background workload, BFQ guarantees to applications about the same responsiveness as if the disk was idle [17].

B. CFQ

CFQ grants disk access to each application for a fixed *time slice*, and schedules slices in a round-robin fashion. Unfortunately, as shown by Valente and Andreolini [17], this service scheme may suffer from both unfairness in throughput distribution and high worst-case delay in request completion time with respect to an ideal, perfectly fair system. In particular, because of these issues and of how the low-latency heuristics work in CFQ, the latter happens to guarantee a worse responsiveness than BFQ [17]. This fact is highlighted also by the results reported in this paper.

III. Missing link for preserving responsiveness

We highlight the problem through a simple example. Consider a system running a guest operating system, say guest G, in a virtual machine, and suppose that either BFQ or CFQ is the default I/O scheduler both in the host and in guest G. Suppose now that a new application, say application A, is being started (loaded) in guest G while other applications are already performing I/O without interruption in the same guest. In these conditions, the cumulative I/O request pattern of guest G, as seen from the host side, may exhibit no special property that allows the BFQ or CFQ scheduler in the host to realize that an application is being loaded in the guest.

Hence, the scheduler in the host may have no reason for privileging the I/O requests coming from guest G. In the end, if also other guests or applications of any other kind are performing I/O in the host—and for the same storage device as guest G—then guest G may receive *no help* to get a high-enough fraction of the disk throughput to start application A quickly. As a conclusion, the start-up time of the application may be high. This is exactly the scenario that we investigate in our experiments. Finally, it is also important to note that our focus has been in local disk/storage, as scheduling of network-based storage systems is not always under the direct control of the Linux scheduling policies.

IV. Extension of the benchmark suite

To implement our experiments we used a publicly available benchmark suite [16] for the Linux operating system. This suite is designed to measure the performance of a disk scheduler with real-world applications. Among the figures of merit measured by the suite, the following two performance indexes are of interest for our experiments:

Aggregate disk throughput. To be of practical interest, a scheduler must guarantee, whenever possible, a high (aggregate) disk throughput. The suite contains a benchmark that allows the disk throughput to be measured

TABLE I. Storage devices used in the experiments

Type	Name	Size	Read peak rate
1.8-inch Hard Disk	Toshiba MK6006GAH	60 GB	10.0 MB/s
microSDHC Card	Transcend SDHC Class 6	8 GB	16 MB/s
eMMC	SanDisk SEM16G	16 GB	70 MB/s

while executing workloads made of the reading and/or the writing of multiple files at the same time.

Responsiveness. Another benchmark of the suite measures the *start-up* time of an application—i.e., how long it takes from when an application is launched to when the application is ready for input—with cold caches and in presence of additional heavy workloads. This time is, in general, a measure of the responsiveness that can be guaranteed to applications in the worst conditions.

Being this benchmark suite designed only for non-virtualized environments, we enabled the above two benchmarks to work correctly also inside a virtual machine, by providing them with the following extensions:

Choice of the disk scheduler in the host. Not only the active disk scheduler in a guest operating system, hereafter abbreviated as just guest OS, is relevant for the I/O performance in the guest itself, but, of course, also the active disk scheduler in the host OS. We extended the benchmarks so as to choose also the latter scheduler.

Host-cache flushing. As a further subtlety, even if the disk cache of the guest OS is empty, the throughput may be however extremely high, and latencies may be extremely low, in the guest OS, if the zone of the guest virtual disk interested by the I/O corresponds to a zone of the host disk already cached in the host OS. To address this issue, and avoid deceptive measurements, we extended both benchmarks to flush caches at the beginning of their execution and, for the responsiveness benchmark, also (just before) each time the application at hand is started. In fact the application is started for a configurable number of times, see Section V.

Workload start and stop in the host. Of course, responsiveness results now depend also on the workload in execution in the host. Actually, the scenario where the responsiveness in a Virtual Machine (VM) is to be carefully evaluated, is exactly the one where the host disk is serving not only the I/O requests arriving from the VM, but also other requests (in fact this is the case that differs most from executing an OS in a non-virtualized environment). We extended the benchmarks to start the desired number of file reads and/or writes also in the host OS. Of course, the benchmarks also automatically shut down the host workload when they finish.

V. Experimental results

We executed our experiments on a Samsung Chromebook, equipped with an ARMv7-A Cortex-A15 (dual-core, 1.7 GHz), 2 GB of RAM and the devices reported in Table I. There was only one VM in execution, hereafter denoted as just *the* VM, emulated using QEMU/KVM. Both the host and the guest OSes were Linux 3.12.

A. Scenarios and measured quantities

We measured, first, the aggregate throughput in the VM while one of the following combinations of workloads was being served.

In the guest. One of the following six workloads, where the tag **type** can be either **seq** or **rand**, with **seq/rand** meaning that files are read or written sequentially/at random positions:

1r-type one reader (i.e., one file being read);
5r-type five parallel readers;
2r2w-type two parallel readers, plus two parallel writers.

In the host. One of the following three workloads (in addition to that generated, in the host, by the VM):

no-host_workload no additional workload in the host;
1r-on_host one sequential file reader in the host;
5r-on_host five sequential parallel readers in the host.

We considered only sequential readers as additional workload in the host, because it was enough to cause the important responsiveness problems shown in our results. In addition, for each workload combination, we repeated the experiments with each of the four possible combinations of active schedulers, choosing between BFQ and CFQ, in the host and in the guest.

The main purpose of the throughput experiments was to verify that in a virtualized environment both schedulers achieved a high-enough throughput to be of practical interest. Both schedulers did achieve, in the guest, about the same (good) performance as in the host. For space limitations, we do not report these results, and focus instead on the main quantity of interest for this paper. In this regard, we measured the start-up time of three popular interactive applications of different sizes, inside the VM and while one of the above combinations of workloads was being served.

The applications were, in increasing-size order: *bash*, the Bourne Again shell, *xterm*, the standard terminal emulator for the X Window System, and *konsole*, the terminal emulator for the K Desktop Environment. As shown by Valente and Andreolini [17], these applications allow their start-up time to be easily computed. In particular, to get worst-case start-up times, we dropped caches both in the guest and in the host before each invocation (Section IV). Finally, just before each invocation a timer was started: if more than 60 seconds elapsed before the application start-up was completed, then the experiment was aborted (as 60 seconds is evidently an unbearable waiting time for an interactive application).

We found that the problem that we want to show, i.e., that responsiveness guarantees are violated in a VM, occurs regardless of which scheduler is used in the host. Besides, in presence of file writers, results are dominated by fluctuations and anomalies caused by the Linux write-back mechanism. These anomalies are almost completely out of the control of the disk schedulers, and not related with the problem that we want to highlight. In the end, we report our detailed results **only with file readers**, **only with BFQ** as the active disk scheduler **in the host**, and **for *xterm***.

B. Statistics details

For each workload combination, we started the application at hand five times, and computed the following statistics over

the measured start-up times: minimum, maximum, average, standard deviation and 95% confidence interval (actually we measured also several other interesting quantities, but in this paper we focus only on application responsiveness). We denote as a *single run* any of these sequences of five invocations. We repeated each single run ten times, and computed the same five statistics as above also across the average start-up times computed for each repetition. We did not find any relevant outlier, hence, for brevity and ease of presentation, in the next plots we show only averages across runs (i.e., averages of the averages computed in each run).

C. Results

Figure 1 shows our results with the hard disk (Table I). The reference line represents the time needed to start *xterm* if the disk is idle, i.e., the minimum possible time that it takes to start *xterm* (a little less than 2 seconds). Comparing this value with the start-up time guaranteed by BFQ with no host workload, and with any of the first three workloads in the guest (first bar for any of the *1r-seq*, *5r-seq* and *1r-rand* guest workloads), we see that, with all these workloads, BFQ guarantees about the same responsiveness as if the disk was idle. The start-up time guaranteed by BFQ is slightly higher with *5r-rand*, for issues related, mainly, to the slightly coarse time granularity guaranteed to scheduled events in the kernel in an ARM embedded system, and to the fact that the reference time itself may advance haltingly in a QEMU VM.

In contrast, again with no host workload, the start-up time guaranteed by CFQ with *1r-seq* or *1r-rand* on the guest is 3 times as high than on an idle disk, whereas with *5r-seq* the start-up time becomes about 17 times as high. With *5r-seq* the figure reports instead an **X** for the start-up time of CFQ: we use this symbolism to indicate that the experiment failed, i.e., that the application did not succeed at all in starting before the 60-second timeout.

In view of the problem highlighted in Section III, the critical scenarios are however the ones with some additional workload in the host; in particular, *1r_on_host* and *5r_on_host* in our experiments. In these scenarios, both schedulers unavoidably fail to preserve a low start-up time. Even with just *1r_on_host*, the start-up time, with BFQ, ranges from 3 to 5.5 times as high than on an idle disk. The start-up time with CFQ is much higher than with BFQ with *1r_on_host* and *1r-seq* on the guest, and, still with *1r_on_host* (and CFQ), is even higher than 60 seconds with *5r-seq* or *5r-rand* on the guest. With *5r_on_host* the start-up time is instead basically unbearable, or even higher than 60 seconds, with both schedulers. Finally, with *1r-rand* all start-up times are lower and more even than with the other guest workloads, because both schedulers do not privilege much random readers, and the background workload is generated by only one reader.

Figures 2 and 3 show our results with the two flash-based devices. At different scales, the patterns are still about the same as with the hard disk. The most notable differences are related to CFQ: on one side, with no additional host workload, CFQ achieves a slightly better performance than on the hard disk, whereas, on the opposite side, CFQ suffers from a much higher degradation of the performance, again with respect to the hard-disk case, in presence of additional host workloads.

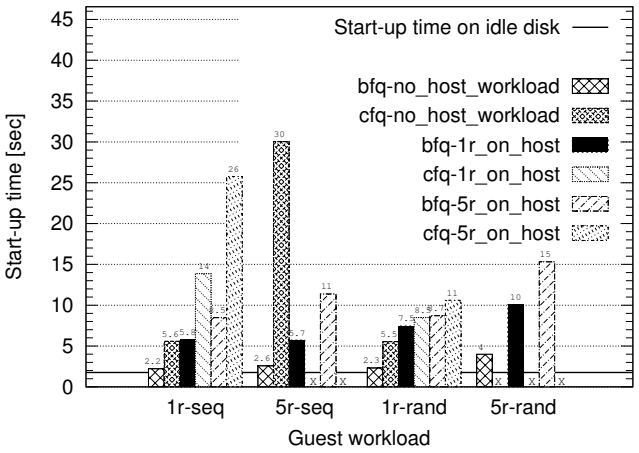


Figure 1. Results with the hard disk (lower is better).

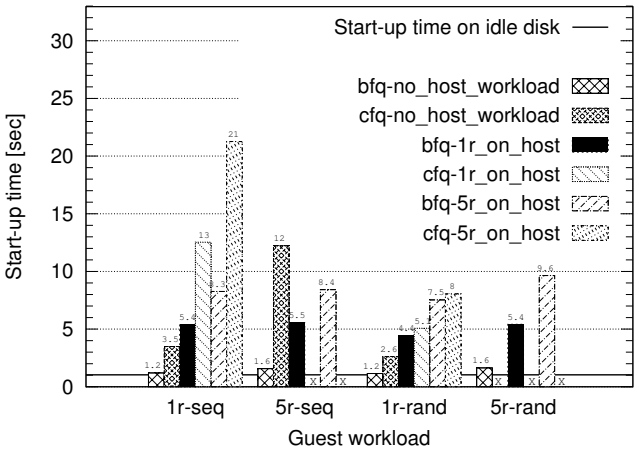


Figure 2. Results with the microSDHC CARD (lower is better).

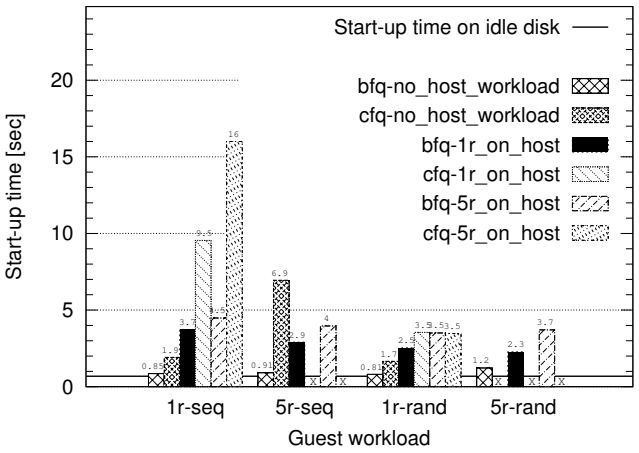


Figure 3. Results with the eMMC (lower is better).

To sum up, our results confirm that, with any of the devices considered, responsiveness guarantees are lost when there is some additional I/O workload in the host.

VI. Possible solutions

The key idea to recover responsiveness is the coordination between schedulers, in order to create the missing link described in Section III. That is why, to deal with the above problem, the guest disk scheduler should somehow inform the host disk scheduler that the I/O requests of the guest should be privileged to preserve a low latency. On the opposite side, the host disk scheduler should properly privilege a guest asking for an urgent and high-throughput access to the disk. In other words, the guest and the host disk schedulers should somehow coordinate with each other to achieve the desired latency goals. As shown in the preceding section, BFQ guarantees a much higher responsiveness than CFQ. For this reason we choose it as the candidate scheduler to extend. The idea is then to realize a coordinated version of BFQ, which we named Virtual-BFQ (V-BFQ). We can describe this idea as follows:

Guest side: when the guest V-BFQ scheduler detects, through its internal heuristics, that some application needs urgent service from the virtual disk, it communicates this need to the V-BFQ scheduler in the host. On the other hand, the V-BFQ guest scheduler also communicates to the host when there are no more applications needing a quick service.

Host side: when the host V-BFQ scheduler receives the above help request from one VM, it privileges that VM until the same VM tells the host V-BFQ scheduler that no help is needed anymore.

From this scheme we can easily deduce that the communication between the guest and host V-BFQ schedulers is a critical issue. Actually, the possible solution space stems mainly from the choices we can make in terms of communication between the schedulers.

We can consider two main alternative approaches to let the guest scheduler communicate to the host scheduler when it needs to be privileged:

A. Augment metadata in guest storage-I/O requests.

We could add some additional boolean field the description of an I/O request, in both the host and the guest kernels. This flag could be set by the guest scheduler for each request coming from an application to be served quickly. The information that this flag is set for a given guest I/O request should then somehow flow through the chain of components that translate I/O requests coming from the guest into corresponding host-side I/O requests. Then the same flag should be set also in the latter requests. Finally, when the host scheduler would see this flag set for an I/O request, it could privilege the host process that generated that requests.

The main benefit of this solution is that it is very simple and little invasive in terms of in-kernel modifications: just the declaration of a data structure should be modified.

There are however two main disadvantages:

Higher latency: host I/O threads would not be privileged immediately (i.e., right after a guest starts to need help), but only when the flagged I/O requests would be eventually issued by these threads. This may be a serious problem with synchronous I/O threads, which issue their next request or batch of requests only after the last pending one has been completed. Until the pending requests of a non-yet-privileged I/O thread are completed, that thread may not issue the flagged one. And, exactly because the thread is not yet privileged, its pending requests may wait for a long time before being served.

User-space modifications would be needed, to let the flag percolate from the I/O requests in the guest kernel to the I/O requests in the host kernel. In this regard, it is worth highlighting an additional important issue: guest I/O requests are currently turned into just simple read-/write operations on the image file of the virtual disk, and not directly into I/O requests. Hence, the chain of components involved in carrying information about this flag, and eventually flagging I/O requests on the host side may be rather long.

B. Immediately signal the need for help to the host with some form of direct communication

For this approach, we can consider two alternative solutions:

B.1 User-space solution: host-side I/O threads serving guest I/O requests of (in-guest) applications needing urgent service may directly ask for their weight to be increased. The main benefit of this solution is that the interaction scheme would be very simple: no modification would be needed in the host scheduler to guess what processes/threads need to be privileged. But there is a significant drawback : it is user-space invasive, QEMU code would need to be modified.

B.2 In-kernel solution: the guest disk scheduler could just tell directly to the host disk scheduler that it needs help to guarantee a high responsiveness to some application. So, no user-space modification needed. But a non-trivial logic would be needed in the host disk scheduler to retrieve the IDs of all the I/O threads to privilege after receiving the help request from the guest. In fact, in QEMU, the threads handling virtual CPUs in a VM differ from the I/O threads that take care of serving I/O requests for that VM. Even worse, I/O threads are dynamically created and destroyed as needed during the lifetime of a VM.

Analyzing the above solution space, we concluded that solution B.2 is the one with the better trade-offs between advantages and disadvantages, and hence decided to extend BFQ accordingly. We provide full details on the resulting implementation, named V-BFQ, in the following section.

VII. The Virtual-BFQ solution

In this section we describe V-BFQ in detail, using pseudo-code. In particular, we focus mostly on the logical aspects.

The first issue to address was how to let the guest V-BFQ communicate directly with the host V-BFQ. As a simple solution, we opted for the Hypervisor-Call instruction available on ARM architectures, hereafter abbreviated as just *hvc*. The execution of an *hvc* generates a Hypervisor Call exception. In particular, if the *hvc* is executed by a KVM/QEMU guest, then a dedicated KVM handler gets called.

Finally, *hvc* has an integer number as an immediate argument. We used two possible values for the immediate argument to let the guest scheduler tell the host scheduler whether it

```
// just after any point in the code where
     *raised_busy_queues* is increased
if (raised_busy_queues == 1) // transition from
     0 to 1
   hvc #1 ; // notify the host that this guest
        needs to be privileged
// just after any point in the code where
     *raised_busy_queues* is decreased
if (raised_busy_queues == 0) // transition from
     1 to 0
   hvc #0 ; // notify the host that this guest
        does not need help anymore
// at the exit from the scheduler
if (raised_busy_queues > 0) // the guest is
     still being privileged
   hvc #0 ; // notify the host that this guest
        does not need help anymore
```

Figure 4. HVC call in the guest

needs to be prioritized or not. In particular, we decided to use the following two values:

1 to indicate the guest needs to be privileged.
0 to indicate the guest does not need to be privileged anymore.

We can now describe in detail both the guest and the host extensions that we integrated in BFQ to implement V-BFQ.

A. Guest extensions

Under Linux, and, in particular, from the BFQ standpoint, a thread is just a process. BFQ basically associates a queue to one or more processes/threads, and raises the weight of the queues associated to processes/threads to be privileged. As a consequence, a guest needs to be privileged if and only if the number of weight-raised and backlogged (i.e., non-empty) queues in the guest BFQ scheduler is higher than 0. In fact, even if there are weight-raised queues, but they are all empty, there is no urgent pending I/O request, and hence there is no need to privilege the guest for the moment.

Fortunately, BFQ maintains a variable that contains exactly the number of backlogged and weight-raised queues. This variable is called *raised_busy_queues* in the code [16]. Hence, to decide when it is time to either ask the host V-BFQ scheduler for a higher fraction of the disk throughput or inform the V-BFQ scheduler that no special treatment is needed anymore, it is enough to track the transitions of this variable, respectively, from 0 to 1 and from 1 to 0. The guest extension of BFQ does exactly that, by invoking an *hvc* with the right argument for each of the two cases. We describe this extension in more detail in the pseudo-code snippet in Figure 4.

B. Host extensions

The service of the I/O requests generated from a guest is delegated by QEMU to a pool of I/O threads created on demand. It follows that:

The V-BFQ scheduler in a host must raise the weights of the queues associated to the I/O threads of a VM whose guest is requesting to be privileged. As a consequence, every time the host V-BFQ scheduler has the opportunity to raise the weight of some queue, it must know what is the set of I/O threads to privilege, so as to check whether that queue is actually associated to one of such threads (and hence must be weight-raised).

As for knowing, every time this information is needed, what is the set of I/O threads to privilege, we need to consider the following important issues.

Although QEMU creates and destroys *supporting* threads, such as I/O threads, dynamically, the group leader of these threads *never changes* for a given VM. Since QEMU creates and kills I/O threads dynamically, when an *hvc #1* is received from a guest, the I/O thread that will handle the guest I/O request that caused the guest to issue that *hvc* may even not yet exist. And the I/O thread group for a VM may change over time, without the V-BFQ scheduler in the host receiving any notification about changes of in set of I/O threads for any VM.

In view of the above issues related to the dynamic creation/destruction of I/O threads, and exploiting the fact that the group leader for a VM is however constant during the lifetime of the VM, we use the following strategy to allow the V-BFQ host scheduler to correctly weight-raise the right queues.

As a basic step, the V-BFQ scheduler maintains a list of (only) the leaders of the groups of I/O threads to be privileged. In more detail, V-BFQ maintains the list of the Process Identifiers (PID) of these leaders. In this respect, it is worth recalling that a thread basically coincides with a process under Linux. From the PID of a group leader it is then extremely simple to scan the list of its current child threads. In particular, the latter list is trivially kept up-to-date by the kernel itself. Hence, when V-BFQ has to decide whether a given queue is to be weight-raised, it consults this list to reconstruct the list of all the threads to privilege.

We describe the host extension of BFQ by describing in detail each of the above two points.

1) Manipulating the list of group leaders: This list of group leaders is updated according to the *hvc #1* or *hvc #0* received from active guests (and also automatically pruned when some group leader is discovered to be non-existing anymore, as shown in detail in Section VII-B2). To achieve this goal, we had first to modify the KVM handler of *hvc* exceptions in the host kernel. The modification are described, using pseudo-code, in Figure 5.

The last function invoked by the *hvc* handler is the V-BFQ hook that handles the update of the list of the PIDs of the leaders of the threads to privilege. Of course, we deduce that, with respect to BFQ, V-BFQ must contain this additional hook. The exact steps made by this hook are described in the Figure 6 where the list of the PIDs of the leaders of the groups of threads to be privileged is named *leader_pid_list*.

As shown in the snippet Figure 6, the hook does not only update the list of group-leader PIDs: if the mode is add the hook also immediately raises, by calling the function *weight_raise_queue()* described in the next section, the weight of all the backlogged queues associated either to the group leader being added or to any of its children. We describe this part and the rest of the host extensions to BFQ in the next

```
// input: data structure describing the qemu
    virtual cpu on which the hvc is executed
HVC-handler(in: vcpu) {
  // get the descriptor of the host-side qemu
      process/thread implementing the vcpu
  qemu_task = get_pid_task(vcpu->pid);
  leader = qemu_task->group_leader; // pid of
      the thread-group leader

  // value passed to hvc when invoked in the
      guest
  arg_value = vcpu->arch.fault.hsr & 0x000000ff;

  if (arg_value == 1)
    mode = add;
  else
    mode = remove;

  // update the list of the leaders of the
      threads to privilege
  V-BFQ-VM_threads_update_hook(leader, mode);
}
```

Figure 5. Pseudo-code of the HVC handler

```
V-BFQ-VM_threads_update_hook(leader, mode) { //
    mode can be either add or remove
  if (mode == add) {
    if (look_for_pid(leader_pid_list) == NULL)
        // pid not present
      add_to_list(leader_pid_list, leader->pid);
  } else {// mode == remove
    pid_entry = look_for_pid(leader_pid_list,
        leader->pid);
    if (pid_entry != NULL) // pid in list
      rm_from_list(leader_pid_list, pid_entry);
  }
  // additional code to achieve maximum
      responsiveness (see below)
  for_each_child_thread(leader)
  // at each iteration child is set to one of
      the child threads
    for_each_backlogged_bfq_queue()
    // at each iteration, bfqq is one of the
        backlogged queue
    if (child->pid == bfqq->pid)
      // to achieve maximum responsiveness,
          immediately raise the weight of
      // the queue and reschedule the queue
          (see below)
      weight_raise_queue(bfqq);
}
```

Figure 6. Pseudo-code of the V-BFQ hook function

section.

2) Raising the weight of the queues associated with the threads to privilege: In the V-BFQ hook, the weights of all the backlogged queues associated either to the group leader being added or to any of its children are raised immediately, because this step is crucial for starting to serve as soon as possible the

```
weight_raise_queue(bfqq) {
  list_entry pid_entry =
      look_for_pid(leader_pid_list, tmp->pid);
  if (pid_entry != NULL) { // pid in list
    bool need_reposition = bfqq !=
        in_service_queue && !bfqq_is_idle;
    if (is_already_raised(bfqq)) {
      move_forward_raising_start_time(bfqq) ;
          // see below
      return; // nothing else to do
    }

  if (need_reposition)
    deactivate(bfqq) ; // remove queue from
        schedule
  perform_core_weight_raising_operations(bfqq);
  if (need_reposition)
    activate(bfqq); // reschedule in the right
        position for the new weight
  }
}

perform_core_weight_raising_operations(bfqq) {
  raise_weight_coeff(bfqq); // raise queue
      weight
  set_raising_start(); // (re)set the start
      time of the weight-raising (see below)
  set_raising_duration(); // (re)set the
      duration of the raising period (see below)
}
```

Figure 7. Pseudo-code of function *weight_raise_queue()*

I/O requests related to a guest asking to be privileged. This step is however effective only provided that the following issue is properly addressed.

Unless it is currently in service, a backlogged queue is of course scheduled for service. In this respect, if the weight of an already-scheduled queue is raised, but the schedule is not changed immediately, then the queue will wait to be served according to its old, low weight. Only after being served, and if still backlogged, the queue will be rescheduled according to its new high weight, and hence, only from that moment on, the queue will get a high fraction of the disk throughput (until its weight is lowered again). In view of this important issue, in the hook, the queues whose weights are raised are also immediately *rescheduled* according to their new weights. This guarantees the minimum possible latency for a guest asking to be privileged. The experimental results in Section VIII clearly show the benefits of this immediate reschedule.

The exact steps taken by the function *weight_raise_queue()* are described in the Figure 7.

As for the functions *move_forward_raising_start_time()*, *set_raising_start()*, and *set_raising_duration()*, these functions are related to how the weight-raising heuristics work in BFQ, and hence in V-BFQ: when weight-raising starts for a queue, BFQ stores the time instant when it happens in a variable that we call *raising_start_time* hereafter. BFQ also sets the duration for the weight-raising: if the queue is constantly backlogged for all this time period, then its weight is lowered again.

In particular, if the queue is already being weight-raised, then V-BFQ just moves forwards its *raising_start_time* , as if the weight-raising period for the queue just (re)started. In fact, differently from the physiological BFQ behavior, for a queue associated to a thread to be privileged, weight-raising is never stopped. To correctly guarantee this special treatment, V-BFQ also contains the following modification with respect to BFQ: at any point in the code where raising might finish for a queue, V-BFQ controls whether that queue is associated to a thread to be privileged, and, in that case, *does not* stop weight-raising for the queue.

To sum up, the part of the host extension of V-BFQ shown so far guarantees that backlogged queues associated to threads to be privileged are immediately weight-raised and, if needed, rescheduled to guarantee minimum latency.

Hence, to cover all possible cases, we are left with handling the case of the queues that: 1) are still idle or not-yet-existing when the hook is invoked to add a new group leader, but 2) are actually associated or, when they become backlogged, will be associated to one of the threads in the group of the just-added leader.

Each of these queues then moves from idle, or non-existing, to backlogged when its first request eventually arrives. At that time, V-BFQ can easily raise the weight of the queue without even needing complex rescheduling operations, because the queue is of course not yet scheduled for service. In this respect, there is however a last subtlety to consider. I/O threads naturally tend to perform random I/O. In fact, even if the original I/O pattern in a guest is sequential, QEMU spawns several I/O threads and each I/O thread will happen to read or write only a chunk of the whole I/O to perform. The merge of these chunks covers a contiguous portion of the virtual disk, but, served separately by each I/O thread, these chunks happens to be located at random positions on the virtual disk.

To still achieve a high throughput also in the presence of this *fragmented* I/O, BFQ merges queues when it detects that they are associated to I/O threads whose merged I/O pattern would be sequential. In particular, this queue merging is realized by choosing a candidate shared queue and redirecting requests arriving from all the I/O threads to the same shared queue. Such a shared queue preserves its original association with a PID. Hence, it may happen that requests coming from I/O threads with a different PID than that stored in a shared queue are however redirected to the queue. In the end, when a new request arrives, to check whether the destination queue is to be weight-raised, it is the PID of the thread making the request to be checked, and not the PID associated to the destination queue.

The steps needed to perform the above control are reported in the function described with pseudo-code in Figure 8. Note that this function also takes care of properly pruning the *leader_pid_list* if needed.

We have now all the elements we need to describe the proper way to extend the BFQ hook, *insert_request()*, invoked to add a new request to a queue, so as to weight-raise a queue associated to an I/O thread to be privileged, when the queue moves from idle or non-existing to backlogged. This function is described in Figure 9.

```
privileged_thread(pid) {
  for_each_element(leader_pid_list) {
  // at each iteration. leader is set to one of
      the elems
    if (thread_no_more_existing(leader)) //
        thread is dead
    rm_from_list(leader_pid_list, leader); //
        remove from list

  for_each_child_thread(leader)
  // at each iteration child is set to one of
      the child threads
    if (pid == child->pid)
      return true ;
  }
  return false ;
}
```

Figure 8. Pseudo-code of the function *privileged_thread*

```
insert_request(request) {
  // rest of the code of the function
  bfqq = get_destination_queue(request);
  if (was_idle_or_nonexisting(bfqq) &&
      privileged_thread(current_thread->pid))
    if (is_already_raised(bfqq))
      move_forward_raising_start_time(bfqq) ;
    else
      perform_core_weight_raising_operations(bfqq)
          ;

  // rest of the code of the function
}
```

Figure 9. Pseudo-code of the function *insert_request()*

There is a final, important issue to consider: a request arrival can be intercepted even before the function *insert_request()* is invoked. In fact a preliminary BFQ hook is invoked just after a thread has obtained an I/O request from the pool of available requests, and has initialized the fields of the requests. In this hook, BFQ inspects the request, and from this inspection it may discover that: 1) the request comes from a thread whose requests are being redirected to a shared queue, but 2) this redirection is not needed anymore (see the code of BFQ for details [16]). If this happens, the code-path that will then be followed in the function *insert_request()* does not pass through the extension described in the code snippet Figure 9. A special *split* portion of the code of the *insert_request()* function is instead executed, to redirect again the requests coming from that thread to the original queue. In this code-path a *resume_state()* function is called to correctly resume the state of this original queue. Accordingly, to properly handle weight-raising for privileged threads also in this special case, we added the code shown in Figure 9 also to the function *resume_state()*.

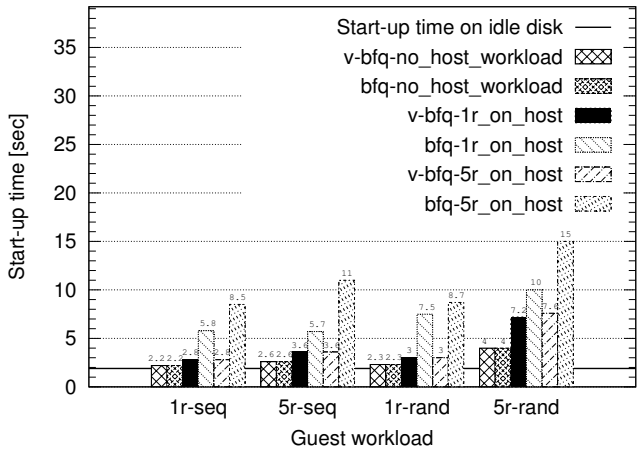


Figure 10. Results with the hard disk and V-BFQ as disk scheduler in both the guest and the host, compared against BFQ as disk scheduler in both the guest and the host (lower is better).

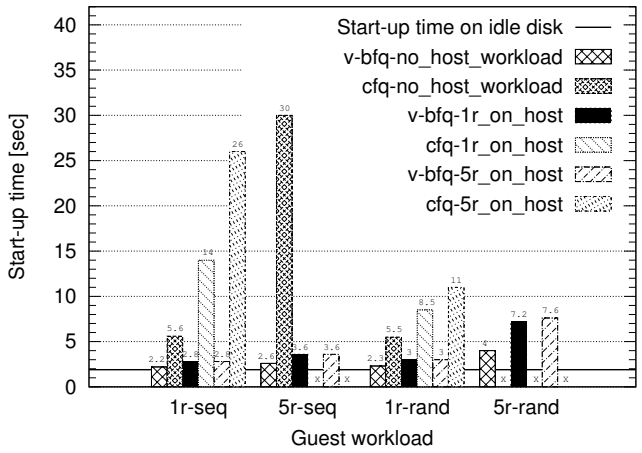


Figure 11. Results with the hard disk and V-BFQ as disk scheduler in both the guest and the host, compared against CFQ as disk scheduler in the guest and BFQ as disk scheduler in the host (lower is better)

VIII. Experimental results with V-BFQ

We repeated the same experiments as in Section V. In particular, as for throughput, V-BFQ trivially achieved the same performance as BFQ, which, in its turn, achieved optimal performance. Hence, for brevity, in this document we do not report throughput results for V-BFQ. Along the same line, we dot not report results for the workloads for which the actual service received by applications has not much to do with the decisions made by the disk schedulers, namely workloads containing greedy writers. And for the scenario where CFQ is used as disk scheduler in the host, because results do not vary significantly depending on whether BFQ or CFQ is used in the host.

A. Results with the hard disk

Figure 10 shows the start-up time recorded in case V-BFQ is used in both the guest and the host. As a reference, in the figure these results are compared against the ones achieved in case BFQ is used in both the guest and the host.

The effectiveness of V-BFQ is evident with *1r-seq*, *5r-seq* and *1r-rand*: regardless of the host workload, with 1r-seq V-BFQ guarantees about the same start-up time as if both the virtual and physical disk were idle. Even with *5r-seq* and *1r-seq*, start-up times are comparable to those recorded when both the virtual and the physical disk are idle.

Start-up times are sensitive to the host workloads with *5r-rand*. In fact, with these workload the issues already highlighted in Section V-C interfere with the correct operation of the heuristics in both the host and the guest V-BFQ schedulers.

To compare the responsiveness achieved by V-BFQ against the one experienced with a typical Linux disk-scheduling configuration, in Figure 11 we compare the start-up times achieved by V-BFQ (i.e., the same values already reported in Figure 10) against the ones recorded when CFQ, i.e., the default Linux I/O scheduler, is used as disk scheduler in the guest. As in Section V, the symbol **X** is used to indicate that the experiment failed because the application did not start within a 60-second timeout. The figure clearly shows the remarkable benefits provided by V-BFQ.

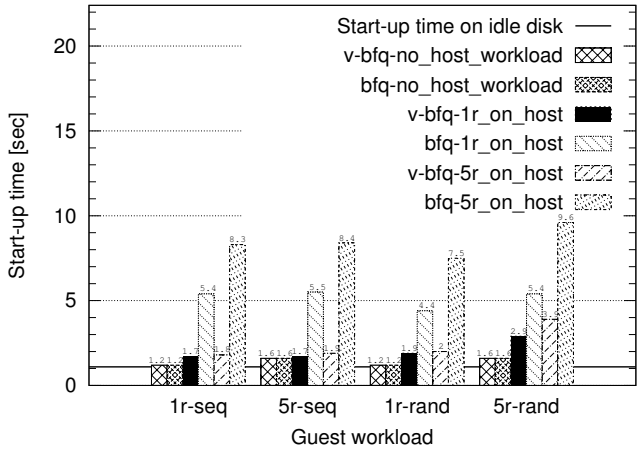


Figure 12. Results with the microSDHC Card and V-BFQ as disk scheduler in both the guest and the host, compared against BFQ as disk scheduler in both the guest and the host (lower is better).

B. Results with the microSDHC Card

As shown in Figures 12 and 13, with the microSDHC Card, results are along the same line as with the hard disk.

C. Results with the eMMC

Finally, also with the eMMC, V-BFQ achieved the same near-optimal performance as with the other two storage devices.

IX. Other schedulers

We also compared V-BFQ with two other schedulers for Linux: Deadline and NOOP in order to point out that V-BFQ is more responsive than all standard I/O schedulers for Linux. The scenario of these experiments is the same in Section VIII and therefore the results can be compared. Only the results for sequential readers are reported.

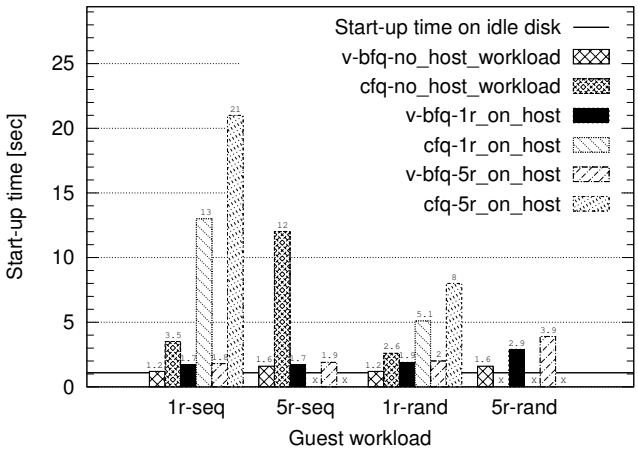


Figure 13. Results with the microSDHC Card and V-BFQ as disk scheduler in both the guest and the host, compared against CFQ as disk scheduler in the guest and BFQ as disk scheduler in the host (lower is better)

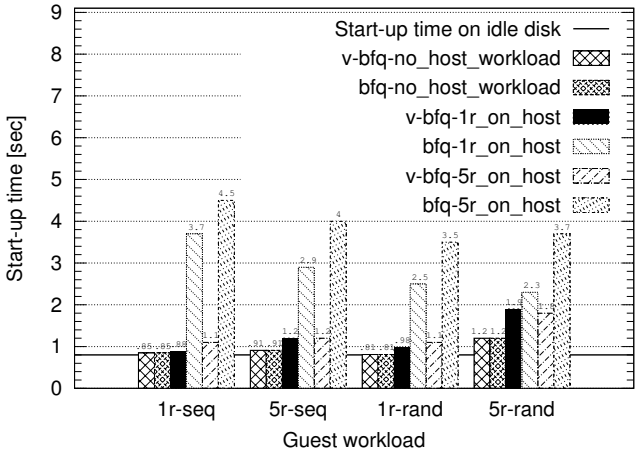


Figure 14. Results with the eMMC and V-BFQ as disk scheduler in both the guest and the host, compared against BFQ as disk scheduler in both the guest and the host (lower is better)

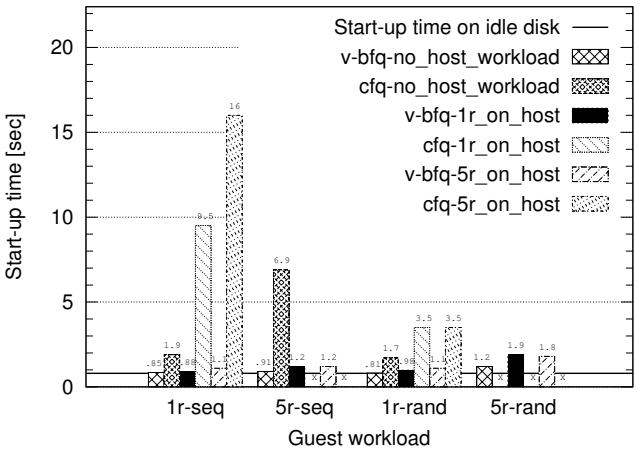


Figure 15. Results with the eMMC and V-BFQ as disk scheduler in both the guest and the host, compared against CFQ as disk scheduler in the guest and BFQ as disk scheduler in the host (lower is better).

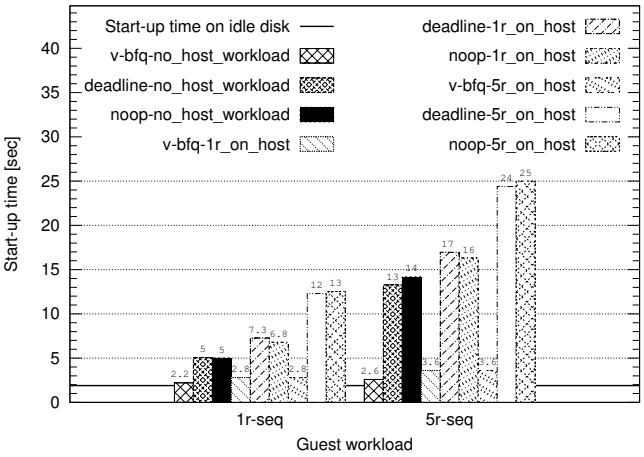


Figure 16. Results with the hard disk and V-BFQ as disk scheduler in both the guest and the host, compared against Deadline and NOOP as disk scheduler in the guest and BFQ as disk scheduler in the host (lower is better).

As it can be seen in Figures 16, 17 and 18 with Deadline or NOOP scheduler in the guest, the start-up time (of *xterm* application) is better than with CFQ as guest scheduler (Figures 11, 13, and 15). And the start-up time for Deadline and NOOP scheduler are roughly equivalent whatever the medium used and the workload. But the latency with V-BFQ as guest scheduler is always lower.

X. Conclusion and Future Work

In this paper, we have shown both theoretically and experimentally that responsiveness guarantees, as related to storage I/O, may be violated in virtualized environments. Even with schedulers, which target to achieve low latency through heuristics, the problem of low responsiveness still persists in virtual machines. The host receives a mix of interactive and background workloads from the guest, which can completely contradict per process heuristics by schedulers such as BFQ. That is why, we have devised a solution, based on BFQ, for preserving responsiveness also in virtualized environments, specifically for embedded systems and the KVM on ARM hypervisor: V-BFQ. This solution introduces the concept of coordinated scheduling between the host/guest scheduler and KVM itself. V-BFQ lived up to its expected performance improvements, guaranteeing high application responsiveness in a virtualized environment, also in the presence of heavy background workloads in both the guest and the host virtual and physical storage devices. Besides, the general scheme adopted to define V-BFQ from BFQ shall be extended and applied also to schedulers of other, important resources, such as CPUs and transmission links. We also plan to extend our investigation to latency guarantees for soft real-time applications (such as audio and video players), and to consider more complex scenarios, such as more than one VM competing for the storage device.

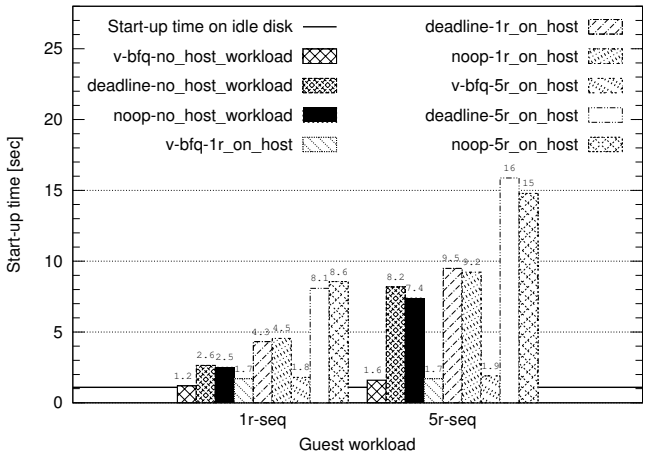


Figure 17. Results with the microSDHC and V-BFQ as disk scheduler in both the guest and the host, compared against Deadline and NOOP as disk scheduler in the guest and BFQ as disk scheduler in the host (lower is better).

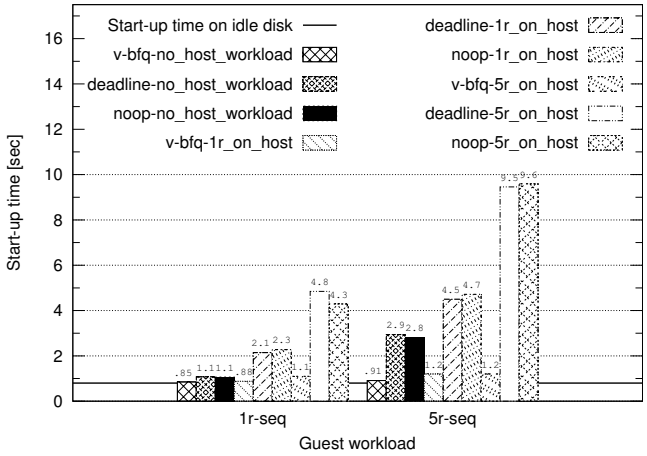


Figure 18. Results with the eMMC and V-BFQ as disk scheduler in both the guest and the host, compared against Deadline and NOOP as disk scheduler in the guest and BFQ as disk scheduler in the host (lower is better).

ACKNOWLEDGMENTS

This research work has been supported by the SeventhFramework Programme (FP7/2007-2013) of the European Community under the grant agreement no. 610640 for the DREAMS project. The authors would like to thank Paolo Valente for providing details about BFQ and his support for the benchmark suite. We also thank the anonymous reviewers for their precious feedback and comments on the manuscript.

REFERENCES

[1] A. Spyridakis and D. Raho, "On Application Responsiveness and Storage Latency in Virtualized Environments," in CLOUD COMPUTING 2014, The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization, 2014, pp. 26-30.

[2] Storage I/O Control Technical Overview [retrieved: November, 2014]. http://www.vmware.com/files/pdf/techpaper/VMW-vSphere41-SIOC.pdf

[3] Virtual disk QoS settings in XenEnterprise [retrieved: November, 2014]. http://docs.vmd.citrix.com/XenServer/4.0.1/reference/ch04s02.html

[4] M. Kesavan, A. Gavrilovska, and K. Schwan,"On disk I/O scheduling in virtual machines," in Proceedings of the 2nd conference on I/O virtualization, USENIX Association, 2010, p. 6.

[5] J. Shafer, "I/O virtualization bottlenecks in cloud computing today," Proceedings of the 2nd conference on I/O virtualization. USENIX Association, 2010, p. 5.

[6] D. Boutcher and A. Chandra, "Does virtualization make disk scheduling passé?," ACM SIGOPS Operating Systems Review 44.1, 2010, pp. 20-24.

[7] X. Pu, L. Liu, Y. Mei, S. Sivathanu, Y. Koh, and C. Pu, "Understanding performance interference of i/o workload in virtualized cloud environments," in Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on, IEEE, 2010, pp. 51-58.

[8] M. Xavier, M. Neves, F. Rossi, T. Ferreto, T. Lange, and C. De Rose, "Performance evaluation of container-based virtualization for high performance computing environments," in Parallel, Distributed and Network-Based Processing (PDP), 2013 21st Euromicro International Conference on, IEEE, 2013, pp. 233-240.

[9] J. Lee et al., "Realizing compositional scheduling through virtualization," IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'12), April 2012, pp. 237-246.

[10] K. Sandstrom, A. Vulgarakis, M. Lindgren, and T. Nolte, "Virtualization technologies in embedded real-time systems," Emerging Technologies & Factory Automation (ETFA), 2013 IEEE 18th Conference on, Sept. 2013, pp. 1-8.

[11] Z. Gu and Q. Zhao, "A state-of-the-art survey on real-time issues in embedded systems virtualization," Journal of Software Engineering and Applications, Vol. 5 No. 4, 2012, pp. 277-290.

[12] H. Kim, H. Lim, J. Jeong, H. Jo, and J. Lee, "Task-aware virtual machine scheduling for I/O performance," in Proceedings of the 2009 ACM SIGPLAN/SIGOPS international conference on Virtual execution environments, ACM, 2009, pp. 101-110.

[13] X. Ling, H. Jin, S. Ibrahim, W. Cao, and S. Wu, "Efficient disk I/O scheduling with QoS guarantee for xen-based hosting platforms," in Cluster, Cloud and Grid Computing (CCGrid), 2012 12th IEEE/ACM International Symposium on, IEEE, 2012, pp. 81-89.

[14] F. Checconi and P. Valente, "High throughput disk scheduling hwith fair bandwidth distribution," IEEE Transactions on Computers, vol. 59, no. 9, May 2010, pp. 1172-1186.

[15] CFQ I/O Scheduler [retrieved: November, 2014]. http://lca2007.linux.org.au/talk/123.html

[16] BFQ homepage [retrieved: November, 2014]. http://algo.ing.unimo.it/people/paolo/disk_sched/

[17] P. Valente and M. Andreolini, "Improving application responsiveness with the BFQ disk I/O scheduler," Proceedings of the 5th Annual International Systems and Storage Conference (SYSTOR '12), June 2012, p. 6.

[18] Virtual-BFQ homepage [retrieved: November, 2014]. http://www.virtualopensystems.com/en/products/virtual-bfq/

An Analytic Evaluation of the SaCS Pattern Language for Conceptualisation of Safety Critical Systems

André Alexandersen Hauge
Institute for Energy Technology, Halden, Norway
andre.hauge@hrp.no

Ketil Stølen
SINTEF ICT, Oslo, Norway
University of Oslo, Norway
ketil.stolen@sintef.no

***Abstract*—In this paper, we present the Safe Control Systems (SaCS) pattern language for the development of conceptual safety designs and conduct an analytical evaluation of the appropriateness of the language for its intended task. By a conceptual safety design we mean an early stage specification of system requirements, system design, and safety case for a safety critical system. The SaCS pattern language can express basic patterns on different aspects of relevance for conceptual safety designs. SaCS can also be used to combine basic patterns into composite patterns. A composite pattern can be instantiated into a conceptual safety design. A framework for evaluating modelling languages is used to conduct the evaluation. The quality of a language is within the framework expressed by six appropriateness factors. A set of requirements is associated with each appropriateness factor. The extent to which these requirements are fulfilled are used to judge the quality. We discuss the fulfilment of the requirements formulated for the language on the basis of the theoretical, technical, and practical considerations that were taken into account and shaped the SaCS language.**

Keywords–pattern language; evaluation; design; conceptualisation; safety.

I. INTRODUCTION

This paper presents the Safe Control Systems (SaCS) pattern language and an evaluation of the suitability of the language as support for the development of safety critical systems. A shorter version of this paper is presented in [1].

A pattern describes a particular recurring problem that arises in a specific context and presents a well-proven generic scheme for its solution [2]. A pattern language is a language for specifying patterns making use of patterns from a vocabulary of existing patterns and defined rules for combining these [3]. The SaCS pattern language has been designed to facilitate the specification of patterns to support the development of conceptual safety designs. With a conceptual safety design, we mean an early stage specification of system requirements, system design, and safety case for a safety critical system. A safety critical system [4] is a system *"whose failure could result in loss of life, significant property damage, or damage to the environment"*. The intended users of the SaCS pattern language are system engineers, safety engineers, hardware and software engineers.

According to McGrath [5], there are eight common methods for evaluation. However, there is no single evaluation method that provides results that are valid across populations (strong on generality), provides very precise measurements (strong on precision), and at the same time is performed in environments that are very similar to reality (strong on realism). Based on the strengths and weaknesses of the different evaluation methods and the questions that are required to be answered, the researcher has to choose how different kinds of methods should be combined. It is desirable to maximise precision, realism, and generality simultaneously but, as argued by McGrath, this is not possible with one single research method.

The suitability of the SaCS pattern language for its intended task is investigated by complementing kinds of evaluations; two case studies and the analytic evaluation presented in this paper. The two case studies, fully documented in [6] and [7], can be seen as variants of what McGrath terms *field experiment*, a method that scores high on realism. According to Eisenhardt [8], the case study approach is especially appropriate in new topic areas and describes how to build theories from case study research. The analytic evaluation is most closely related to *non-empirical evidence* in the McGrath classification, a method that scores high on generality.

A framework for analysing languages known as the Semiotic Quality (SEQUAL) framework [9] is used as a basis for the analytic evaluation. The appropriateness of a language for its intended task is in the framework characterised by six appropriateness factors [9]: domain, modeller, participant, comprehensibility, tool, and organisational. A set of requirements is presented for each appropriateness factor in order to characterise more precisely what is expected from our language in order to be appropriate. The requirements represent the criteria for judging what is appropriate of a language for conceptual safety design, independent of SaCS being appropriate or not. We motivate our choices and discuss to what extent the requirements are fulfilled.

The remainder of this paper is structured as follows: Section II provides an introduction to the SaCS pattern language. Section III demonstrates the applicability of the SaCS pattern language in an example. Section IV discusses analytic evaluation approaches and motivates the selection of the SEQUAL framework. Section V presents the analytic evaluation of the SaCS pattern language according to the SEQUAL framework. Section VI presents related work on pattern-based development. Section VII draws the conclusions.

II. Background on the SaCS Pattern Language

The SaCS pattern language is an integrated part of the SaCS method. In order explain the language, we outline the SaCS method. The SaCS method consists of the following three artefacts:

A. *The SaCS process*: defines the process for systematically applying SaCS patterns to support the development of conceptual safety designs.

B. *The SaCS library*: defines 26 basic SaCS patterns on best practices for conceptual safety design categorised into six different kinds. A user defines composite SaCS patterns on the basis of patterns in the library. The user can extend the library by defining additional basic and composite SaCS patterns.

C. *The SaCS pattern language*: defines how to express basic SaCS patterns and includes a graphical notation for specifying composite SaCS patterns.

In order to apply the SaCS method, an assumed user of the SaCS process employs the patterns within the library as guidance to problem solving. Furthermore, the user of the process employs the language for expressing a solution to a problem in the form of a pattern in order to extend the library. The language has a formal syntax as well as a structured semantics [10] that supports users in specifying patterns and understanding what is expressed by a pattern.

Depending on the complexity of the problem that needs to be solved in a given context, and to the extent available patterns can be used to solve the problem, the user chooses whether to address the problem with a single basic pattern or rather by a combination of several patterns. The classification structure for the patterns denotes the different kinds of patterns offered by the library. As a basic pattern provides guidance on a specific problem-solution concept with a limited scope, the use of several and complementary kinds of basic patterns is necessary for conceptual safety design. The combination of several basic patterns for problem solving facilitates separation of concerns.

A composite pattern is expressed graphically and specifies how several patterns are combined. The visual presentation facilitates the discussion between different kinds of users on how conceptual safety design is intended to be approached with patterns as guidance. The instantiation of a composite that combines suitable patterns supporting the specification of requirements, system design, and safety case in a given context produces the conceptual safety design.

In the following sub-sections we briefly describe each of the artefacts that are part of the SaCS method.

A. The SaCS process

The SaCS process interleaves three main activities, each of which is divided into sub-activities:

- *Pattern Selection*: The purpose of this activity is to support the conception of a design by selecting: a) SaCS patterns for requirement elicitation; b) SaCS patterns for establishing design basis; and c) SaCS patterns for establishing safety case.
- *Pattern Composition*: The purpose of this activity is to specify the use of the selected patterns by specifying: a) compositions of patterns; and b) instantiations of patterns.
- *Pattern Instantiation*: The purpose of this activity is to instantiate the composite pattern specification by: a) selecting pattern instantiation order; and b) conducting stepwise instantiation.

The SaCS process is exemplified in Section III. The example shows how a pattern selection map is used as support for pattern selection, and how the pattern language supports pattern composition. Pattern instantiation is supported by instantiation rules defined for every basic pattern (fully defined in [6][7]). The definition of one of the basic patterns in SaCS is also presented in Section II-B.

B. The SaCS Library

The SaCS library consists of a set of basic patterns as well as the composite patterns defined by a user on the basis of pre-defined patterns. While a basic pattern is defined by text and illustrations, a composite is defined graphically. In the following, a slightly formatted version of the basic pattern *Establish System Safety Requirements* documented in [6] is reproduced as an example of the content of the library. The pattern is described as a sequence of named sections. The section names are presented in a **bold** font. The section named "Pattern Signature" contains an illustration classifying the pattern as well as its inputs and output parameters according to the syntax of the SaCS pattern langauge [10]. The section "Process Solution" contains a UML activity diagram with SaCS language specific annotations, presented in Fig. 2. The SaCS specific annotations are represented by: the dotted drawn frame that encapsulates the activity diagram; the dotted drawn boxes that appears on the dotted drawn frame; and the arrows that are connected to the dotted drawn boxes. The dotted drawn boxes represent either an input or an output. The identifier within a box names a parameter. An arrow pointing away from a box indicates that the identified parameter is an input. An arrow pointing towards a box indicates that the identified parameter is an output. The remaining diagram elements represent the process solution in terms of a UML activity diagram. The SaCS specific annotations in Fig. 2 indicate in what way inputs are related to the different activities of the process for establishing safety requirements and how the result of one of the activities represents an output of applying the pattern.

Name: Establish System Safety Requirements

Pattern Signature: *Establish System Safety Requirements* is defined with the signature illustrated in Fig. 1. In Fig. 1, the following abbreviations are used for denoting the parameters of the pattern:

- *ToA* is short for Target of Assessment.
- *Reg* is short for Regulations.
- *Risks* is not abbreviated; represents the documentation of the risks associated with the application of *ToA* in its intended context.
- *Req* is short for Requirements.

Figure 1. Establish System Safety Requirements – Pattern Signature

Intent: Support the specification of system safety requirements *Req* on the basis of a risk-based approach. The safety requirements describe the required measures to be satisfied by the system *ToA* to assure the necessary safety integrity. The general approach for defining safety requirements is to define them on the basis of the result of a risk assessment *Risks*, especially the mitigations identified as means to reduce risk to an acceptable level. The pattern describes the general process of capturing the requirements that must be satisfied in order to assure safety.

Applicability: The *Establish System Safety Requirements* pattern is intended for the following situations:

- When the system under construction can negatively affect the overall system safety.
- When there are identified measures that can mitigate identified risks and can be used as input to the specification of safety requirements.

Problem: The main aspects relevant to address when establishing the safety requirements are:

- *Characteristics*: To define the system characteristics to be satisfied such that the occurrence of unwanted events are minimised or avoided.
- *Functions*: To define the safety functions that assures safe operations.
- *Constraints*: To define the functional constrains that sufficiently delimit potentially hazardous operations.
- *Environment*: To define the operational environment that ensures safe operations.
- *Compliance*: To define the requirements that are required to be satisfied in order to comply with laws, regulation, and standards, as a minimum the mandatory requirements related to assurance of safety. These requirements include requirements on applying some specific development process, performing certain activities, or making use of specific techniques.

Process Solution: Fig. 2 illustrates the *Establish System Safety Requirements* process specified using a UML activity diagram.

The input parameters associated with the activity diagram can be interpreted as follows:

- *ToA* (Target of Assessment): represents the target system for which safety requirements should be established.
- *Reg* (Regulations): represents any source of information describing mandatory or recommended practices (e.g. as provided in laws, regulations or standards) valuable for identifying risk reducing measures.
- *Risks*: represents risks associated with the target system.

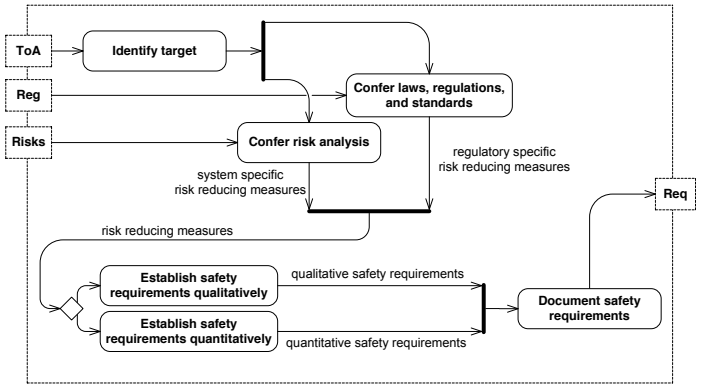


Figure 2. Establish System Safety Requirements – Process Flow

The main activities serve the following purpose:

- *Identify target*: the intent of the activity is to identify *ToA*. The description of the target should as a minimum include a definition of the system and its boundaries, its operational profile, functional requirements, and safety integrity requirements.
- *Confer laws, regulations, and standards*: the intent of the activity is to capture all relevant data (requirements for risk reducing measures) from relevant sources (normative references) in order to outline the set of risk reducing measures that shall be met by compliance. Each source is inspected in order to identify, as a minimum, the mandatory risk reducing measures that shall be met in order to be compliant.
- *Confer risk analysis*: the intent of the activity is to capture all the relevant data on risk analysis of the system that is under construction in order to outline the system specific risk reducing measures that shall be met.
- *Establish safety requirements qualitatively*: the intent of the activity is to define safety requirements on the basis of those identified risk reducing measures required applied, and which can be demonstrated fulfilled with qualitative reasoning.
- *Establish safety requirements quantitatively*: the intent of the activity is to define safety requirements on the basis of those identified risk reducing measures required applied, and which can be demonstrated fulfilled with quantitative reasoning.
- *Document safety requirements*: the intent of the activity is to detail all relevant information with respect to the requirements in a system safety requirements specification. For each requirement defined in the requirement specification, information detailing what influenced its definition should be provided, e.g., the

associated risks, assumptions, calculations, and justifications.

Instantiation Rule: An artefact *Req* (see Fig. 1 and Fig. 2) is the result of a process that instantiates the *Establish System Safety Requirements* pattern if:

- *Req* is a set of requirements.
- *Req* is a result of applying a process illustrated in Fig. 2 and described in Section "Process Solution". The process is initiated by an activity on describing the target *ToA*. Once a description of the target system and its operational context is provided, the next activities shall identify the risk reducing measures to be applied to the target by conferring relevant laws, regulations and standards as well as the result of target specific risk analysis for guidance. Once all the relevant risk-reducing measures are identified, these shall be used as a basis to define the requirements to be met by the target system or by the process to be followed while developing the target. The requirements are defined quantitatively or qualitatively depending on the nature of the risk reducing measure that is addressed. The requirements are documented in a requirement specification *Req*.
- Every requirement of *Req* is traceable to relevant risks (identified by the instantiation of *Risks*), and/or regulatory requirements (identified by the instantiation of *Reg*).
- Every requirement of *Req* is justified such that any assumptions, calculations, and assessments that support the specification of the requirement as a safety requirement are provided.

Related Patterns: The *Establish System Safety Requirements* pattern is related to other patterns in the following manner:

- can succeed the *Risk Analysis* pattern that supports identifying risks. The *Establish System Safety Requirements* can be applied as support for defining the requirements to be fulfilled in order to reduce risk to an acceptable risk level.
- can be used in order to detail requirements for the design that is a result of an instantiation of a design pattern.

C. The SaCS Pattern Language

Fig. 3 defines a composite pattern according to the syntax of SaCS [10]. The composite described in Fig. 3 is named *Safety Requirements* and consists of the basic patterns *Hazard Analysis*, *Risk Analysis*, and *Establish System Safety Requirements*. The contained patterns of *Safety Requirements* are referenced graphically. The basic patterns are specified separately in a structured manner comparable to what can be found in the literature [2][3][11][12][13][14][15][16] on patterns, e.g., as in the case of the pattern *Establish System Safety Requirements* presented in Section II-B.

In Fig. 3, the horizontal line separates the declaration part of the composite pattern from its content. The icon placed below the identifier *Safety Requirements* signals that this is

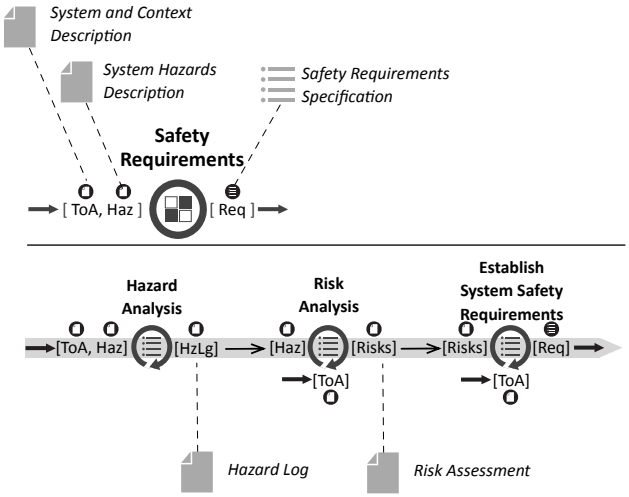


Figure 3. A composite pattern named Safety Requirements

a composite pattern. Every pattern in SaCS is parameterised. An input parameter represents the information expected to be provided when applying a pattern in a context. An output parameter represents the expected outcome of applying a pattern in a given context. The inputs to *Safety Requirements* are listed inside square brackets to the left of the icon, i.e., *ToA* and *Haz*. The arrow pointing towards the brackets symbolises input. The output of the pattern is also listed inside square brackets, but on the right-hand side of the icon, i.e., *Req*. The arrow pointing away from the brackets symbolises output. An icon placed adjacent to a parameter identifier denotes its type. The parameters *ToA*, *Haz*, *HzLg*, and *Risks* in Fig. 3 are of type *documentation*, while *Req* is of type *requirement*. The inputs and outputs of a composite are always publicly accessible.

A particular instantiation of a parameter is documented by a relation that connects a parameter with its associated development artefact. In Fig. 3, a grey icon placed adjacent to an identifier of a development artefact classifies what kind of artefact that is referenced. A dotted drawn line connecting a parameter with an artefact represents an *instantiates* relation. Instantiations of parameters expressed in Fig. 3 are:

- The document artefact *System and Context Description* instantiates *ToA*.
- The document artefact *System Hazards Description* instantiates *Haz*.
- The requirement artefact *Safety Requirements Specification* instantiates *Req*.
- The document artefact *Hazard Log* instantiates *HzLg*.
- The document artefact *Risk Assessment* instantiates *Risks*.

A one-to-many relationship exists between inputs in the declaration part of a composite and similarly named inputs with public accessibility (those pointed at by fat arrows) in the content part. The relationship is such that when *ToA* of *Safety Requirements* is instantiated (i.e., given its value by the defined relation to *System and Context Description*) then every correspondingly named input parameter contained

in the composite is also similarly instantiated. A one-to-one relationship exists between an output parameter in the declaration part of a composite and a correspondingly named output parameter with public accessibility (those followed by a fat arrow) in the content part. The relationship is such that when *Req* of *Establish System Safety Requirements* is produced then *Req* of *Safety Requirements* is similarly produced.

The arrows (thin arrows) connecting basic patterns in the content part of *Safety Requirements* represent two instances of an operator known as the *assigns* relation. The *assigns* relations within *Safety Requirements* express that:

- The output *HzLg* of the pattern *Hazard Analysis* is assigned to the input *Haz* of the pattern *Risk Analysis*.
- The output *Risks* of the pattern *Risk Analysis* is assigned to the input *Risks* of the pattern *Establish System Safety Requirements*.

That the three basic patterns are process patterns follows from the icon below their respective identifiers. There are six different kinds of basic patterns in SaCS, each represented by a specific icon.

The notation for expressing composite patterns consists of the following main modelling elements:

Pattern reference: Fig. 4 presents the icons for the different kinds of patterns defined in SaCS. A pattern reference consists of a unique identifier in a **bold** font and an icon classifying the pattern referenced.

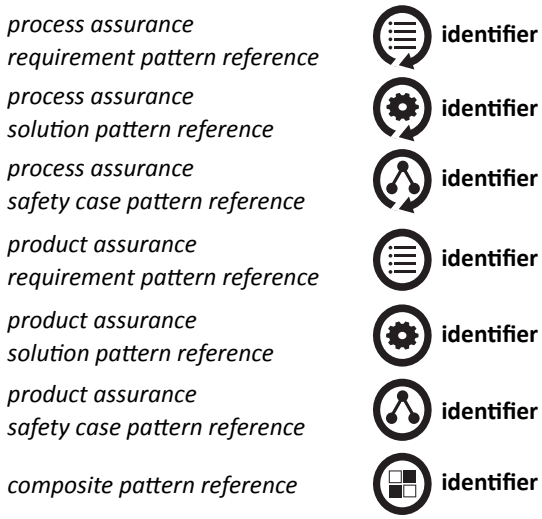


Figure 4. The icons for the different kinds of pattern references in SaCS

Parameter: Fig. 5 presents the icons for the different kinds of parameters defined in SaCS. A parameter consists of an identifier and an icon classifying the parameter. Within a composite, the parameters of a pattern are listed inside square brackets and placed adjacent to the icon that classifies the pattern. The *documentation parameter* is a general classification and represent a parameter that cannot be classified as a *requirement parameter*, *design parameter* or *safety case parameter*.

Relation: Fig. 6 presents the symbols for different kinds of relations defined in SaCS. A relation denotes a relationship

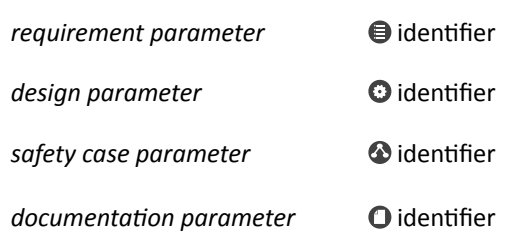


Figure 5. The icons for different kinds of parameters in SaCS

between elements in a composite pattern. The *instantiates* relation is used to associate an artefact with a parameter indicating that the artefact instantiates the parameter. The *assigns* relation models a data flow between patterns where the output of one pattern is used as an input to a second pattern. The *combines* relation is used to denote that the outputs of the patterns that are related are combined into a set consisting of the union of all outputs. The *details* relation is used to denote that an output of a pattern is detailed by the output of a related pattern. The *satisfies* relation is used to denote that an output of a pattern (typically represented by a requirement parameter) is satisfied by the output of a related pattern (typically represented by a design parameter). The *demonstrates* relation is used to denote that an output (typically represented by a safety case parameter) is a safety demonstration for the output of a related pattern (typically represented by a design parameter).

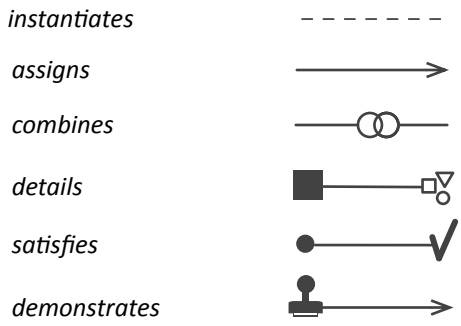


Figure 6. The symbols for the different kinds of relations in SaCS

Artefact reference: Fig. 7 presents the different kinds of artefact references defined in SaCS. An artefact reference consists of an unique identifier in an *italics* font and an icon classifying what kind of artefact that is referenced. Artefact references are used for denoting a specific representation of parameters.

Instantiation order: A composite pattern may include symbols as guidance to the user on the proper instantiation order of patterns. In the serial instantiation case of Fig. 8 there are two composite patterns *A* and *B*, where *A* shall be instantiated before *B*. A general rule is that patterns placed closer to the starting point of the arrow are instantiated prior to patterns placed close to the tip of the arrow. In the parallel instantiation case of Fig. 8, no specific ordering of the patterns *A* and *B* is assumed and thus the respective pattern references are placed on two separate arrows.

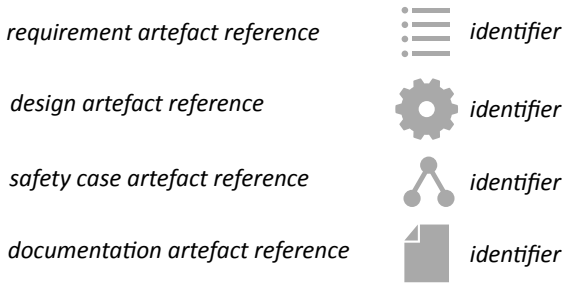


Figure 7. The icons for different kinds of artefact references in SaCS

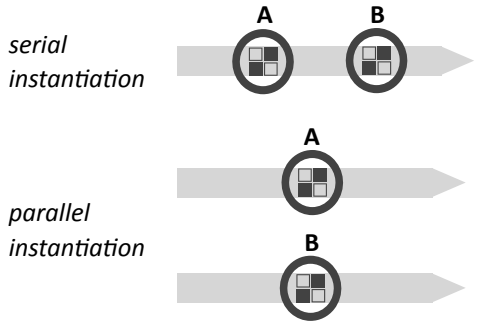


Figure 8. The symbols for the different kinds of instantiation orders in SaCS

III. SaCS EXEMPLIFIED

Fig. 9 exemplifies the integration of the three artefacts into the SaCS method. In Fig. 9, arrows are used to indicate the flow between the use of the SaCS process, the use of the library, and the language as support in performing the activities of the process. Filled arrows indicate the inputs and outputs from the application of the SaCS method. The dotted arrow indicates that a user may choose to feed the composite pattern provided as a result of applying SaCS back into the library of patterns.

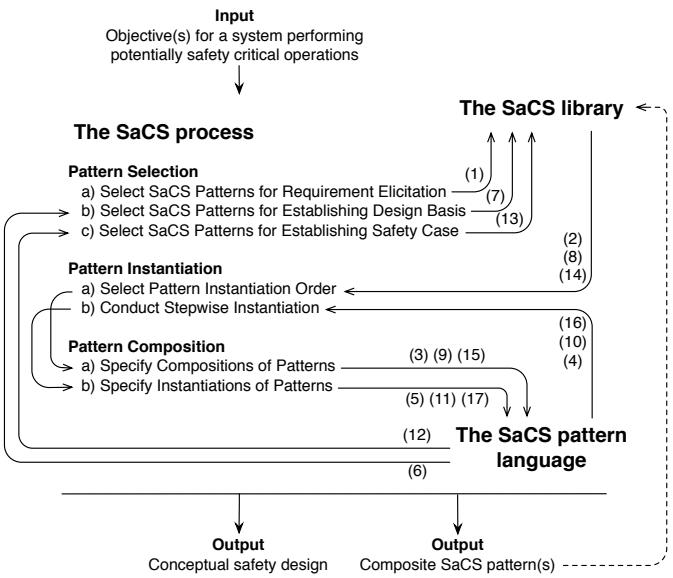


Figure 9. Example integration of the three artefacts into the SaCS method

In the following, the SaCS method is explained by exemplifying the steps (1) to (17) from Fig. 9. The example shows the main steps in the application of the SaCS method for developing a conceptual safety design of a railway interlocking system. A comprehensive explanation can be found in [7]; an outline is given below.

Fig. 10 illustrates a train station with two tracks. The station is connected in both ends of the station area to neighbouring stations with a single track. There are eight train routes possible with the track configuration illustrated in Fig. 10. An interlocking system controls the movements of trains along defined train routes. The interlocking system actuates the different distant signals, main signals, and point equipment according to defined rules in order to enforce safe train movements. A distant signal gives the train driver indication on the signalling to expect from the associated main signal.

Fig. 11 presents references to development documents that are available to an assumed user in the following example. *Interlocking concept description* is a document assumed to define the main functionality of an interlocking system for separating train movements at a train station as illustrated in Fig. 10. Furthermore, *Hazard log* is a document assumed to describe the result from an initial hazard analysis of the interlocking concept. There are patterns in the library that facilitate the definition of an initial concept as well as hazard identification and analysis, but we nevertheless assume the presence of these documents as a starting point in the exemplification of the use of the SaCS method. The end result is expected to be a conceptual safety design of a railway interlocking system.

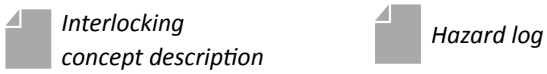


Figure 11. Representation of references to development documentation

Step (1): Fig. 12 illustrates a fragment of a larger pattern selection map with the addition of annotations to indicate in which steps of this example we use the map. A user traverses the pattern selection map in the order indicated by the arrows and considers whether a pattern is relevant for application based on its definition. The diamonds represent choices. The patterns below a diamond represent the alternatives associated with a choice where more than one pattern can be selected.

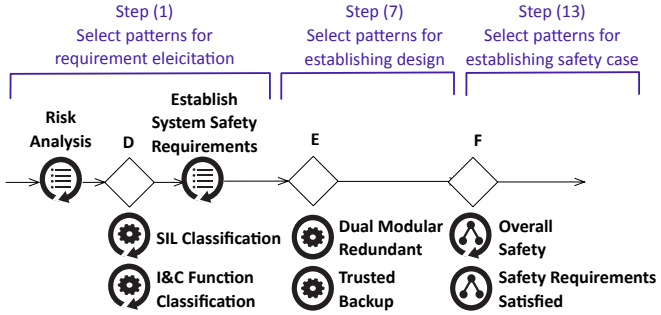


Figure 12. Pattern selection map (simplified)

By the use of Fig. 12, the user starts the pattern selection activity from the left and identify *Risk Analysis* as the first

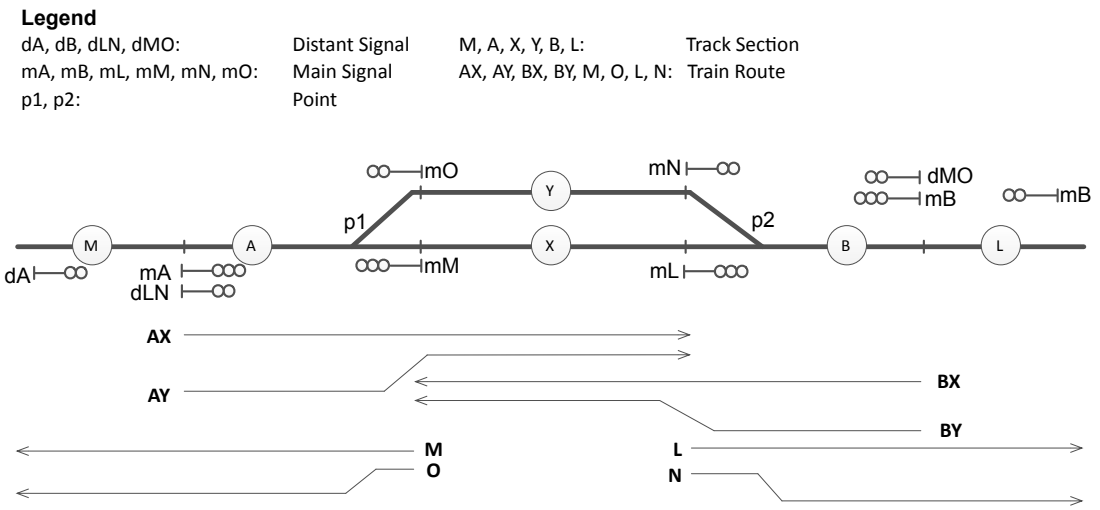


Figure 10. Railway case (adopted from [7])

pattern that should be considered as support. By inspecting the definition of *Risk Analysis*, the user identifies that the pattern describes the process of performing risk analysis on the basis of the results from hazard analysis. As a hazard log is already present, the user selects the pattern as support for the assessment of risks. In choice D, two patterns are identified that offer guidance on alternative methods for the classification of functions, named *SIL Classification* (SIL is short for Safety Integrity Level) and *I&C Function Classification* (I&C is short for Instrumentation and Control), respectively. The user selects *SIL Classification* as support as it defines an approach to the classification of functions commonly applied within the railway domain whereas *I&C Function Classification* is applicable within the nuclear power production domain. The pattern *Establish System Safety Requirements* was presented in detail earlier. The pattern describes how the result from risk analysis should be used as input to the specification of safety requirements. The user selects these three patterns as support for the elicitation of requirements in Step (1). The user will return to the selection map in Step (7) and Step (13) related to the selection of patterns for establishing design and safety case.

Step (2)-(3): Fig. 13 presents how an assumed user combines the three patterns selected in the previous step according to the syntax of the SaCS pattern language. The order in which these patterns should be instantiated is indicated in the pattern selection map presented in Fig. 12. The order can also be found by inspecting the "Related Patterns" sections of the pattern definitions. In Fig. 13, the order is specified by the wide grey arrow in the background indicating that *Risk Analysis* and *SIL Classification* can be instantiated in parallel and prior to *Establish System Safety Requirements*. As described earlier, the symbols [ ] embrace a parameter list. The parameters are abbreviated as follows: *ToA* is short for Target of Assessment, *Haz* is short for Hazards, *Req* is short for Requirements, *FncCat* is short for Function Categorisation, *ClsCr* is short for Classification of Criticality, and *Risks* is not abbreviated. The small icons adjacent to the parameters classify the parameters. The thin arrows represents three instances of an *assigns* relation.

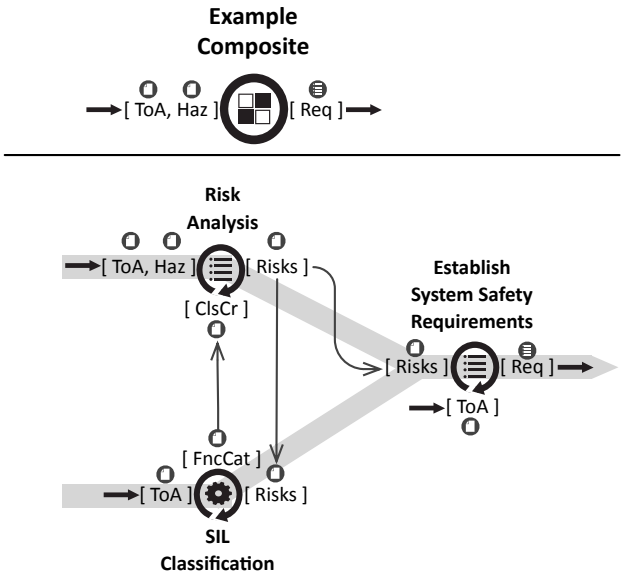


Figure 13. Composite that can be instantiated into a specification of requirements

Step (4)-(5): Fig. 14 is identical to Fig. 13 with the addition of annotations indicating the instantiation of patterns. In Fig. 14, the instantiation of the composite is documented such that *Interlocking concept description* represents the documentation associated with the input parameter *ToA* and *Hazard log* is associated with the input *Haz*. We have assumed that the user has instantiated the composite to produce three different documents. The outcome of instantiating the composite is defined as being represented by a requirements specification known as *Safety requirements specification*. Furthermore, two intermediate documents to the requirements specification is produced where *Classification of functions* is associated with the output parameter *FncCat* of *SIL Classification* and *Risk analysis results* is a document associated with the output *Risks* of *Risk Analysis*. An example finding from the risk analysis is that erroneously positioned points can cause train derailment

or collision. Thus, the interlocking system is clearly a safety critical system. Furthermore, the interlocking system must assure that points are always positioned correctly. An example of a safety requirement that addresses this is *"SR.2: A train route may not be locked unless all points belonging to the train route are positioned correctly."*

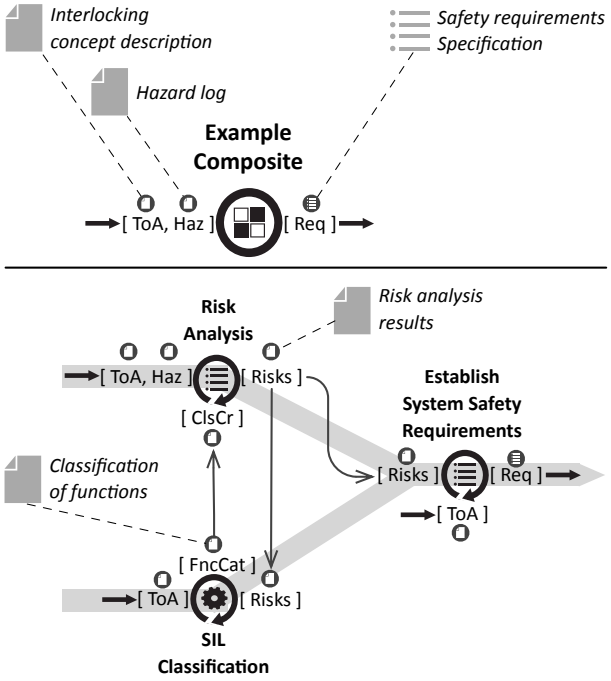


Figure 14. Composite specifying its instantiation into a specification of requirements

Step (6)-(8): Once the user has specified the requirements for the system under construction, enough information should be available for selecting an appropriate design pattern to use as a basis for establishing the system design. Thus, in Step (7) the user continues the pattern selection activity from where it was temporarily stopped in Step (1). In the pattern selection map presented in Fig. 12, we have arrived at choice E. In choice E, the design patterns *Dual Modular Redundant* and *Trusted Backup* are indicated as alternatives. We assume that the user finds *Dual Modular Redundant* to offer the most beneficial design after an evaluation of the ability of the respective design solutions described within these two patterns to satisfy the relevant requirements. The user selects *Dual Modular Redundant* as support for system design.

Step (9)-(11): Fig. 15 is identical to Fig. 14 with the addition of annotations indicating the instantiation of the design pattern named *Dual Modular Redundant*. The detailing of the application of patterns for establishing design basis is the concern of Step (9) and Step (11). However, the actual instantiation of patterns for system design is the concern of Step (10).

In Step (10), the user is supposed to make use of the requirements derived earlier as input for detailing a system design. We assume that the user instantiate *Dual Modular Redundant* according to its instantiation rule to produce a system design specification. Fig. 16 is a simplified UML component diagram showing an excerpt of the system design specification

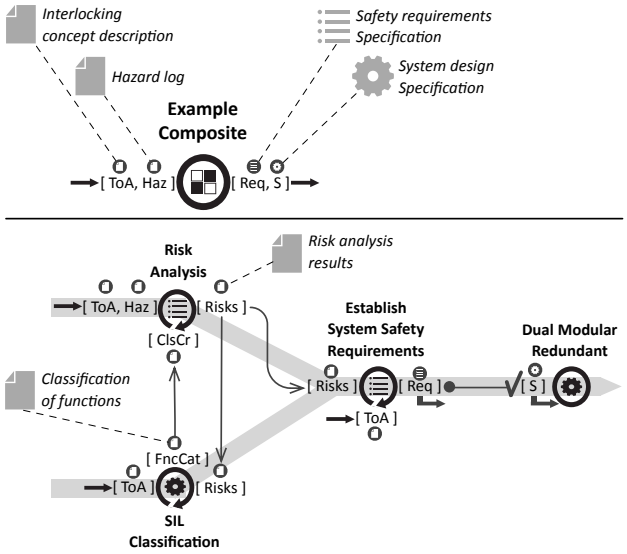


Figure 15. Composite specifying its instantiation into a specification of requirements and system design

that describes the interlocking system. The interlocking system consists of dual controller components (Ctr 1 and Ctr 2) that implements the interlocking rules. The command unit (Cmd) is responsible for handling the interaction with the operator. The interlocking system interacts with equipment like lights, points and train detection equipment through dedicated interfaces. A voter assures fail-safe behaviour (e.g., all lights indicate stop) in the case of disagreement between the redundant controllers.

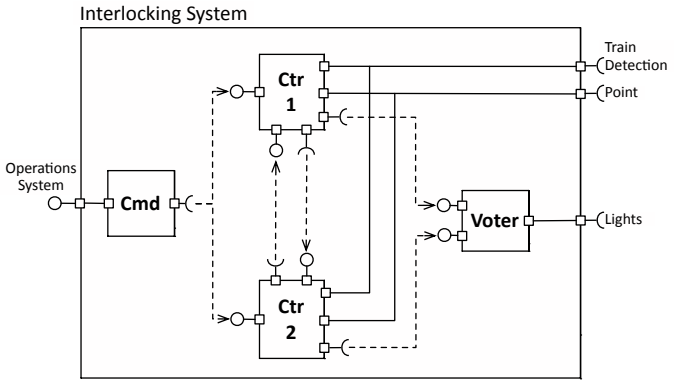


Figure 16. Excerpt of the system design specification – The main components

Fig. 17 is a simplified UML state machine diagram exemplifying the specification of the behaviour of the interlocking system.

The diagram details the interlocking system behaviour given a request for locking a train route AX (see Fig. 10). In Fig. 17, details are only given for the state "Check Points in Correct Position" as this is relevant for the example requirement stated earlier (see SR.2). While Fig. 16 specifies the main components of the system under construction according to the guidance within *Dual Modular Redundant*, Fig. 17 specifies the behaviour of the system in accordance with the requirements derived with the use of *Establish System Safety*

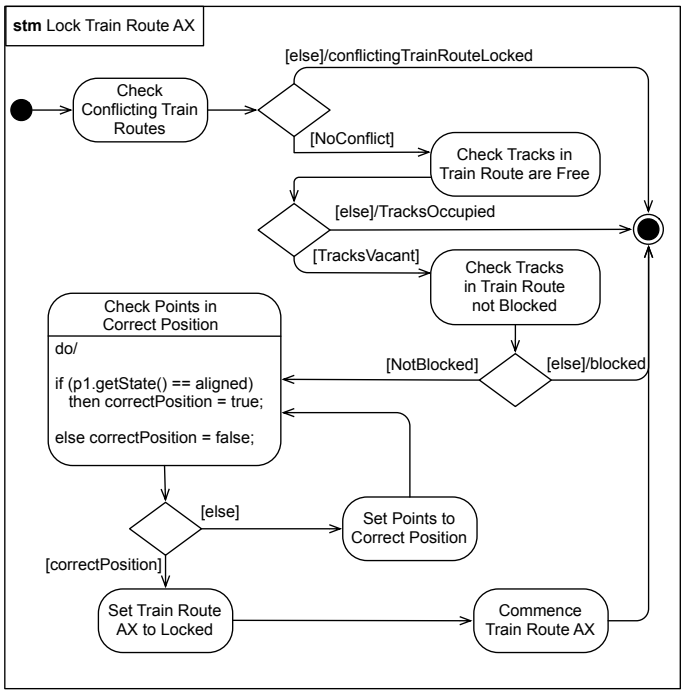


Figure 17. Excerpt of the system design specification – The behaviour exemplified

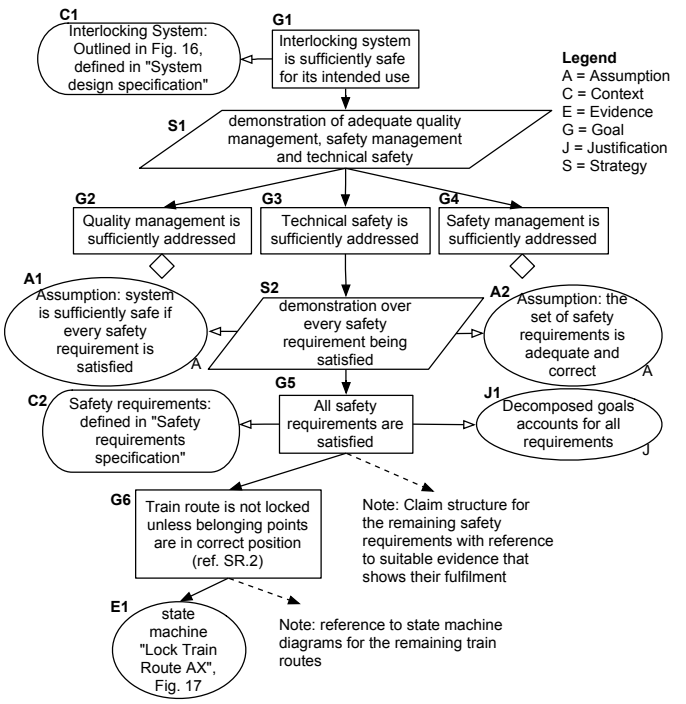


Figure 18. A safety case expressed with the GSN notation [15]

Requirements.

Step (12)-(14): Once the system design is established, enough information should be available for the user to start the detailing of how safety will be argued. Thus, in Step (13) the user continues the pattern selection activity from where it was temporarily stopped in Step (7), leading to choice F in Fig. 12. In choice F, the user inspects the pattern definitions of the different proposed patterns. We assume that the user finds *Overall Safety* as a suitable starting point. The pattern provides guidance on arguing safety from a quality management, safety management, as well as a technical safety perspective. This fits well with the railway standard EN 50129 [17], which requires that these three perspectives are explicitly addressed in a safety case for railway signalling systems.

Step (15)-(16): While we postpone Step (15) for later in order to avoid unnecessary repetitions of similar pattern compositions (the result of the step can be seen in Fig. 19), we assume in Step (16) that the user makes use of the system design derived earlier as a definition of the target for which a safety case shall be defined. We further assume that the user instantiate *Overall Safety* selected in Step (13) according to its instantiation rule to produce a safety case as presented in Fig. 18.

The overall claim (expressed in a goal element) in the safety case presented in Fig. 18 states that the proposed "interlocking system is sufficiently safe for its intended use." The overall claim is decomposed into sub-claims, which at some point is supported by evidence (referenced within an evidence element). A diamond symbolises that the goal is not developed. The claim structure is simplified to only account for requirement SR.2, which was defined earlier. Furthermore, the claim structure is also simplified to only reference evidence for the correct specification of required behaviour as is given within this paper. The relevant specification of behaviour with respect to arguing the fulfilment of requirement SR.2 is defined in Fig. 17.

Step (17): In this step the user specifies the end result of pattern composition. Fig. 19 extends Fig. 15 to also specify the use of the pattern *Overall Safety*. Fig. 19 includes the information intended to be specified at Step (15), which was postponed.

In Fig. 19, the parameters of the patterns introduced are abbreviated as follows: *S* is short for System, *ToD* is short for Target of Demonstration, and *Case* is short for Safety Case. The *satisfies* relation (see Fig. 6) expresses that a design *S* (represented by the *System design specification*) shall satisfy the requirements in *Req* (represented by *Safety requirements specification*). Furthermore, *S* of *Dual Modular Redundant* represents the target of demonstration as defined by the *assigns* relation connecting *S* with *ToD* of *Overall Safety*. The outcome *Case* (represented by *Safety case specification*) of *Overall Safety* is related to *S* of *Dual Modular Redundant* with a *demonstrates* relation. The *demonstrates* relation expresses that *Case* is a safety demonstration for *S*.

In the example, the SaCS method is assumed applied for developing a conceptual safety design of a railway interlocking system. The result is here partly represented by the specifications in Fig. 16, Fig. 17, and Fig. 18. The conceptual safety design in the example is the result of instantiating the composite pattern expressed in Fig. 19. In the composite, the triple that represents the conceptual safety design is assumed represented by the documentations referred to within Fig. 19 as *Safety requirements specification*, *System design specification*, and *Safety case specification*.

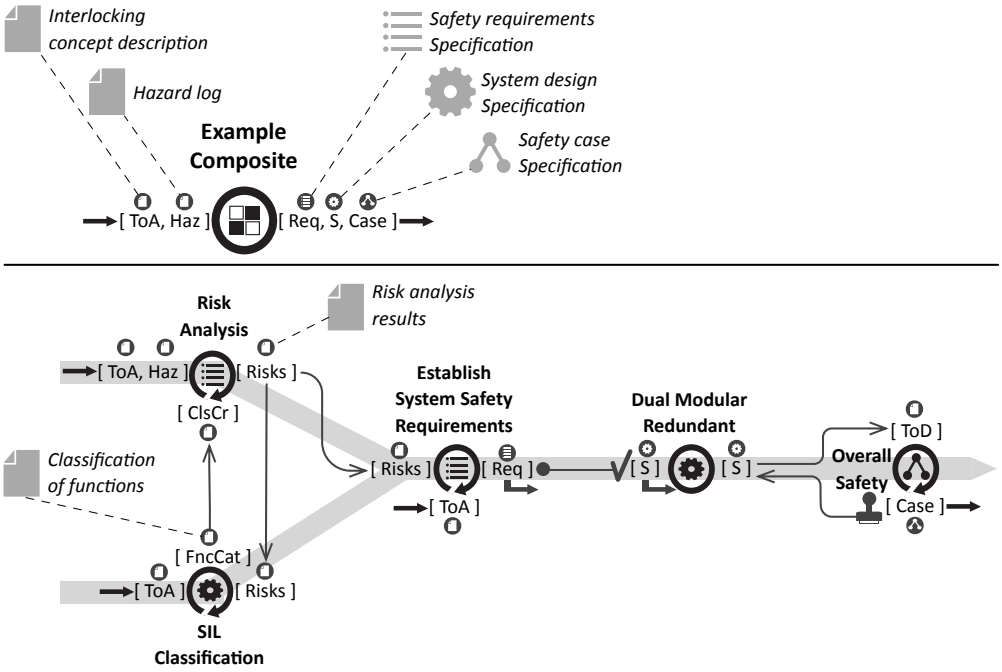


Figure 19. Composite specifying its instantiation into a conceptual safety design

IV. THE FRAMEWORK USED AS SUPPORT FOR THE ANALYTIC EVALUATION

Mendling et al. [18] describe two dominant approaches in the literature for evaluating the quality of modelling approaches: (1) top-down quality frameworks; (2) bottom-up metrics that relate to quality aspects. The most prominent top-down quality framework according to [18] is SEQUAL [9][19][20]. The framework is based on semiotic theory (the theory of signs) and is developed for evaluating the quality of conceptual models and languages of all kinds. Moody et al. [21] report on an empiric study involving 194 participants on the use of SEQUAL and concludes that the study provides strong support for the validity of the framework. Becker et al. [22] present a guideline-based approach as an alternative to SEQUAL. It addresses the six factors: correctness, clarity, relevance, comparability, economic efficiency, and systematic design. Mendling et al. [18] also discuss a number of bottom-up metrics approaches. Several of these contributions are theoretic without empirical validation according to the authors.

We have chosen to apply the SEQUAL framework for our evaluation as it is a general framework applicable to different kinds of languages [9] whose usefulness has been confirmed in experiments [21]. Furthermore, an analytic evaluation is preferred over a metric-based approach due to project limitations. An analytic evaluation is also a suitable complement to the experience-based evaluations of SaCS presented in [6] and [7].

According to SEQUAL, the appropriateness of a modelling language for a specific task is related to the definition of the following sets: the set of goals G for the modelling task; its domain D in the form of the set of all statements that can be stated about the situation at hand; the relevant knowledge of the modeller Km and other participants Ks involved in the modelling task; what persons involved interpret the models to say I; the language L in the form of the set of all statements that can be expressed in the language; relevant tool interpretation T of the models; and what is expressed in the models M.

Fig. 20 is adopted from [23] and illustrates the relationships between the different sets in SEQUAL. The quality of a language L is expressed by six appropriateness factors. The quality of a model M is expressed by nine quality aspects.

In the following, we will not address the different quality aspects of a model M but rather address the quality of the SaCS pattern language.

The appropriateness factors indicated in Fig. 20 are related to different properties of the language under evaluation. The appropriateness factors are [9]:

- *Domain appropriateness*: the language should be able to represent all concepts in the domain.
- *Modeller appropriateness*: there should be no statements in the explicit knowledge of the modeller that cannot be expressed in the language.
- *Participant appropriateness*: the conceptual basis should correspond as much as possible to the way individuals who partake in modelling perceive reality.
- *Comprehensibility appropriateness*: participants in the modelling should be able to understand all the possible statements of the language.
- *Tool appropriateness*: the language should have a syntax and semantics that a computerised tool can understand.
- *Organisational appropriateness*: the language should be usable within the organisation it targets such that

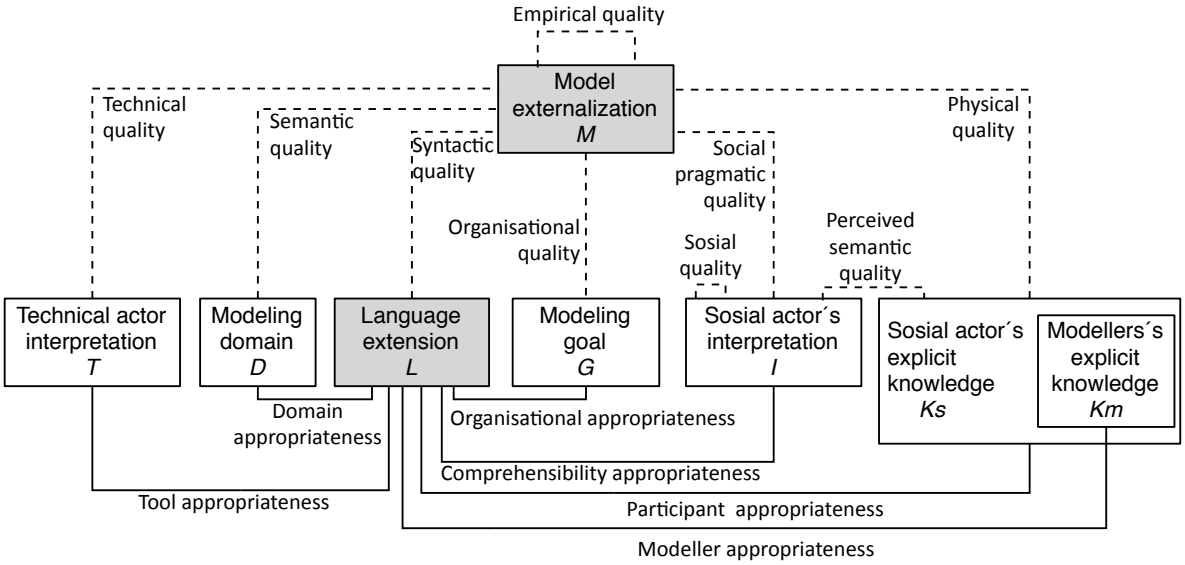


Figure 20. The quality framework (adopted from [23])

it fits with the work processes and the modelling required to be performed.

A set of requirements is associated with each appropriateness factor. The extent to which the requirements are fulfilled are used to judge the quality of the SaCS pattern language for its intended task. The requirements are defined on the basis of requirements found in the literature on SEQUAL.

V. THE ANALYTIC EVALUATION

A necessary step in the application of SEQUAL [9][19][23] is to adapt the evaluation to account for the modelling needs. This amounts to expressing what the different appropriateness factors of the framework represent in the particular context of the evaluation in question. In particular, the modelling needs are detailed by the definition of a set of criteria for each of the appropriateness factors.

Table I introduces the criteria for evaluating the suitability of the SaCS pattern language for its intended task. In the first column of Table I, the two letters of each requirement identifier identify the appropriateness factor addressed by the requirement, e.g., DA for Domain Appropriateness.

The different appropriateness factors are addressed successively in Section V-A to Section V-F according to the order in Table I. Each requirement from Table I is discussed. A requirement identifier is presented in a **bold** font when first introduced in the text followed by the associated requirement and an evaluation of the extent to which the requirement is fulfilled by SaCS.

TABLE I. OVERVIEW OF EVALUATION CRITERIA

ID	Requirement
DA.1	The language must include the concepts representing best practices within conceptual safety design.
DA.2	The language must support the application of best practices within conceptual safety design.
MA.1	The language must facilitate tacit knowledge externalisation within conceptual safety design.
MA.2	The language must support the modelling needs within conceptual safety design.
PA.1	The terms used for concepts in the language must be the same terms used within safety engineering.
PA.2	The symbols used to illustrate the meaning of concepts in the language must reflect these meanings.
PA.3	The language must be understandable for people familiar with safety engineering without specific training.
CA.1	The concepts and symbols of the language should differ to the extent they are different.
CA.2	It must be possible to group related statements in the language in a natural manner.
CA.3	It must be possible to reduce model complexity with the language.
CA.4	The symbols of the language should be as simple as possible with appropriate use of colour and emphasis.
TA.1	The language must have a precise syntax.
TA.2	The language must have a precise semantics.
OA.1	The language must be able to express the desired conceptual safety design when applied in a safety context.
OA.2	The language must ease the comprehensibility of best practices within conceptual safety design for relevant target groups like system engineers, safety engineers, hardware and software engineers.
OA.3	The language must be usable without the need of costly tools.

A. Domain appropriateness

DA.1 The language must include the concepts representing best practices within conceptual safety design.

In the SaCS language, there are currently 26 basic patterns [6][7] on different concepts within conceptual safety design. Each pattern can be referenced by its unique name. The knowledge confined within SaCS patterns is extracted from many sources, but is also, to a large extent, traceable to a few important and influential sources within the field of development of safety critical systems. We regard international safety standards and guidelines as particularly suitable sources of inspiration as these are:

- developed and matured over many years;
- defined on the basis of a consensus between an international group of domain and safety experts;
- defined according to established processes for quality assurance within highly regarded organisations;

- defined with the intention of addressing recurring challenges within development of safety critical systems;
- defined with the intention of describing acceptable solutions for developing safety critical systems.

We find it fair to argue that a pattern that expresses a concept in accordance with highly regarded international safety standards and guidelines or otherwise authoritative documents within a domain, expresses a practice that is commonly accepted. The knowledge captured within the patterns in the library reflects the knowledge within the safety literature in the following manner:

1) *Establish Concept*: The pattern captures the essence of the first phase in the system life-cycle presented in EN 50126 [24] and in IEC 61508 [25] that is simply named "Concept". The phase is in the standards concerned with how to establish the purpose and constraints associated with a system under development.

2) *Hazard Identification*: The pattern describes a process for identifying hazards in accordance with the practise defined in EN 50129 [17]. The pattern captures the hazard identification part of the phase named "Hazard and risk analysis" of the safety life-cycle presented in IEC 61508 [25]. The identification of hazards is essential for later steps concerned with the definition of safety requirements.

3) *Hazard Analysis*: The pattern describes a process for identifying the potential causes of hazards in accordance with the practise defined in EN 50129 [17]. The patterns in 2), 3), and 4) captures the intent expressed in the life-cycle phase named "Hazard and risk analysis" in IEC 61508 [25].

4) *Risk Analysis*: The pattern describes a process for assessing risk in accordance with the practises defined in EN 50129 [17] and IEC 61508 [25].

5) *Establish System Safety Requirements*: The pattern describes a process for specifying safety requirements inspired by the fourth phase in the system life-cycle presented in EN 50126 [24] named "System Requirements". In EN 50129 [17], the safety requirements are defined on the basis of results from hazard identification and analysis, risk assessment, and the classification of functions. Thus, the patterns in 2), 3), 4), 5), and 9) may be used as a set of complementing patterns supporting the elicitation of safety requirements in a manner comparable to the practice described in EN 50129. In a similar manner, the fourth phase of the safety life-cycle presented in IEC 61508 [25] is named 'Overall safety requirements" and represents a phase that is concerned with the specification of requirements on the basis of hazard and risk analysis.

6) *FMEA*: The pattern captures the essence of the Failure Modes and Effects Analysis (FMEA) method. The FMEA method is widely used within domains developing safety critical systems and is described in IEC 60812 [26].

7) *FTA*: The pattern captures the essence of the Fault Tree Analysis (FTA) method as it is described in IEC 61025 [27].

8) *I&C Functions Categorisation*: The pattern captures the method for classifying nuclear Instrumentation and Control (I&C) functions as defined within IEC 61226 [28].

9) *SIL Classification*: The pattern captures the railway approach to the classification of functions as it is defined in EN 50128 [29], applicable for software, and EN 50129 [17], applicable for system functions.

10) *Variable Demand for Service*: The pattern is highly specialised to support requirements elicitation for the conceptualisation of a nuclear control system in a case study described in [6]. The case study describes a very specific development challenge and the pattern describes a solution to that challenge. The pattern is not defined on the basis of the knowledge confined within the safety standards and guidelines literature. It describes, however, a systematic approach for requirements elicitation according to principles for effective requirements engineering. We cannot argue that the pattern describes an effective solution to a recurring challenge within conceptual safety design in general.

11) *Station Interlocking Requirements*: The pattern captures the essential requirements for building interlocking systems as defined by the Norwegian Rail Authority in the technical rules JD 550 [30].

12) *Level Crossing Interlocking Requirements*: The pattern captures the essential requirements for building level crossing systems as defined by the Norwegian Rail Authority in the technical rules JD 550 [30].

13) *Trusted Backup*: The pattern describes a system design concept enabling the utilisation of adaptable control systems for safety critical control tasks by the use of a variant of the Simplex architecture proposed by Sha [31]. Sha refers to the Boeing 777 flight control system as an example of a system that uses the Simplex architecture in practise.

14) *Dual Modular Redundant*: The pattern defines a variant of a generic design solution [32] consisting of two redundant controllers and a voting unit that is implemented in numerous kinds of systems for different kinds of task.

15) *Overall Safety*: The pattern defines a structure for providing an overall system safety demonstration in a manner that is comparable to the overall structure required for safety cases as presented in EN 50129 [17].

16) *Technical Safety*: The pattern describes a structure for arguing safety with a focus on technical aspects and represents a variant of Part 4 of the safety case required by EN 50129 [17] addressing issues related to technical safety.

17) *Code of Practice*: The pattern defines a structure for arguing that safety objectives are met on the basis of the application of well-proven practices. The strategy of arguing safety on the basis of the application

of a code of practice is expressed in the European Regulation on common safety methods within the railway industry [33] and its associated application guideline [34].

18) *Cross Reference* [6]: The pattern describes a structure for arguing that a system satisfies safety objectives on the basis of a comparison between the system in question with a similar and already accepted system. The strategy is expressed in the European Regulation on common safety methods within the railway industry [33] and its associated application guideline [34].

19) *Explicit Risk Evaluation*: The pattern describes a structure for arguing that a target system is sufficiently safe on the basis of risk being sufficiently addressed. The strategy is expressed in the European Regulation on common safety methods within the railway industry [33] and its associated application guideline [34].

20) *Safety Requirements Satisfied*: The pattern describes a structure for arguing that a target system is sufficiently safe on the basis of evidence for safety requirements being satisfied. The practice of demonstrating system safety on the basis of demonstrating that safety requirements are satisfied is one of the core principles of EN 50129 [17] and IEC 61508 [25].

21) *Deterministic Evidence*: The pattern describes an argument structure where a claim is supported by evidence that demonstrates the claim is fully predictable. One example of the need for relying on deterministic evidence is related to the recommendation expressed in Table A.12 within Appendix A of EN 50128 [29] where it is expressed that the software code of SIL 4 systems is expected to contain no dynamic objects, no dynamic variables, and no conditional jumps.

22) *Assessment Evidence*: The pattern describes an argument structure where a claim is supported by evidence derived on the basis of the application of a suitable assessment method. One example of the need to argue that suitable assessment techniques has been applied can be seen in Appendix A of EN 50128 [29]. Appendix A of EN 50128 provides recommendations on the application of specific techniques for assessing software depending on their Software Safety Integrity Level (SWSIL) classification.

23) *Process Quality Evidence*: The pattern describes an argument structure where the evidence of compliance to a particular process, as well as evidences of the quality of the process, assures a claim being met.

24) *Process Compliance Evidence*: The pattern describes an argument structure where the evidence of compliance to a process that is argued widely known as providing effective results assures a claim being met. A typical use of this strategy is to claim that the software in a given system is developed according to the practices described in a relevant software standard (e.g., [29][35]) and thus is developed according to acceptable practices.

25) *Probabilistic Evidence*: The pattern describes an argument structure where the evidence supporting a claim is derived on the basis of probabilistic methods. Appendix A of EN 50128 [29] identifies some of the recommended techniques that may be used to derive probabilistic results such as reliability block diagram, fault tree analysis, and Markov models.

26) *Basic Assumption Evidence*: The pattern describes an argument structure where a claim is supported by a form of rationale or a justified assumption such that no further evidence is required. In any kind of argumentation there are some axioms that are used as base facts. The assumptions used as a basis in assuring and justifying that safety objectives are met should be justified [17][24][25][29][36].

The icons and symbols of the SaCS pattern language is presented in Section II-C. Fig. 4 presents the icons used for SaCS patterns within a composite pattern specification and indicates a categorisation. The first three icons are used for categorising patterns providing development guidance with a strong processual focus. The next three icons are used for categorising patterns providing development guidance with a strong product focus. The last icon is used for categorising composite patterns. Different kinds of patterns express different concepts and best practices within development of safety critical systems. The combined use of patterns from different categories facilitates development of conceptual safety designs.

Habli and Kelly [37] describe the two dominant approaches in safety standards for providing assurance of safety objectives being met. These are: (1) the process-based approach; (2) the product-based approach. Within the process-based approach, safety assurance is achieved on the basis of evidence from the application of recommended or mandatory development practices in the development life cycle. Within the product-based approach, safety assurance is achieved on the basis of product specific evidences that meet safety requirements derived from hazard analysis. The practice within safety standards, as described above, motivates our categorisation into the process assurance and the product assurance pattern groups.

The safety property of a system is addressed on the basis of a demonstration of the fulfilment of safety objectives. Seven nuclear regulators [36] define a safety demonstration as *"a set of arguments and evidence elements that support a selected set of dependability claims - in particular the safety - of the operation of a system important to safety used in a given plant environment"*. Although it is the end system that is put into operation, evidences supporting safety claims are produced throughout the system life cycle and need to be systematically gathered from the very beginning of a development project [36]. The safety case approach represents a means for explicitly presenting the structure of claims, arguments, and evidences in a manner that facilitates evaluation of the rationale and basis for claiming that safety objectives are met. The safety case approach is supported by several authors [15][36][37][38]. What is described above motivates the need for patterns supporting safety case specification in addition to patterns on requirements elicitation and system design specification.

As indicated above, in the design of the SaCS pattern language we have as much as possible described practices, selected keywords, and designed icons in the spirit of leading literature within the area. This indicates that we at least are

able to represent a significant part of the concepts of relevance for conceptual safety design.

DA.2 The language must support the application of best practices within conceptual safety design.

Safety standards, e.g., IEC 61508 [25], typically demand a number of activities to be performed in which certain activities must be applied in a specific sequence. Furthermore, safety standards can also describe the expected inputs and outputs of different activities and in this sense describe what is the expected content of deliverables that allows a transition from one activity to the next. According to Krogstie [9], the main phenomena in languages that accommodate a behavioural modelling perspective are states and transitions between states. In this sense, the language should support the modelling of the application of best practices according to a behavioural modelling perspective.

Fig. 5 presents the icons for the different kinds of parameters and Fig. 7 presents the artefact references in SaCS. The *documentation parameter* and the *documentation artefact reference* types (represented visually by the icons presented in Fig. 5 and Fig. 7) are defined in order to allow a generic classification of parameters and artefacts that cannot be classified as requirement, design, or safety case. An example can be the result of risk analysis that is an intermediate result in conceptual safety design and an input to an activity on the specification of safety requirements [17][25]. The process of deriving safety requirements on the basis of an assessment of hazards is expressed by a chain of patterns as presented in Fig. 3. The outcome of applying the last pattern in the chain is a requirements specification. The last pattern cannot be applied before the required inputs are produced.

Fig. 6 presents the symbolic representation of the different relations in SaCS. Relations define transitions between patterns or dependencies between elements within a composite pattern definition. The reports [6][7] define the concepts behind the different relations and exemplify the practical use of all the concepts in different scenarios. Fig. 3 in Section II exemplifies a composite pattern containing five instances of the *instantiates* relation and two instances of the *assigns* relation.

The need for the different relations presented in Fig. 6 is motivated by the practices described in different standards and guidelines, e.g., IEC 61508 [25], where activities like hazard identification and hazard analysis are required to be performed sequentially and where the output of one activity is assigned as input to another activity. Thus, we need a concept of assignment. In SaCS, this is defined by an *assigns* relation between patterns. When performing an activity like hazard analysis, the results from the application of a number of methods can be combined and used as input. Two widely used methods are captured in two different basic SaCS patterns known as Failure Modes and Effects Analysis (FMEA) and Fault Tree Analysis (FTA). A concept for combining results is needed in order to model that the results from applying several patterns such as FMEA and FTA are combined into a union consisting of all individual results. In SaCS, this is defined by a *combines* relation between patterns. A *details* relation is used to express that the result of applying one pattern is further detailed by the application of a second pattern. Functional safety is an important concept in IEC 61508 [25]. Functional safety is a part of the overall safety that depends on a system or equipment operating correctly in response to its inputs. Furthermore, functional safety is achieved when every specified safety function is carried out and the level of performance required of each safety function is met. A *satisfies* relation between a pattern for requirements elicitation and a pattern for system design expresses that the derived system satisfies the derived requirements. Safety case patterns support documenting the safety argument. A *demonstrates* relation between a safety case pattern and a design pattern expresses that the derived safety argument represents a safety demonstration for the derived system design.

Fig. 8 illustrates how the intended instantiation order of patterns can be visualised. The direction of the arrow indicates the pattern instantiation order; patterns (or more precisely the patterns referred to graphically) placed closer to the starting point of the arrow are instantiated prior to patterns placed close to the tip of the arrow. Patterns can be instantiated in parallel and thus have no specific order; this is visualised by placing pattern references on separate arrows.

As argued above, the SaCS language facilitates the application of best practices within safety design and mirrors leading international standards within the area; in particular IEC 61508. We therefore think it is fair to say that the language to a large extent fulfils DA.2.

B. Modeller appropriateness

MA.1 The language must facilitate tacit knowledge externalisation within conceptual safety design.

As already mentioned, the current version of the language contains 26 basic patterns. The basic patterns are documented in [6] and [7]. The patterns are defined on the basis of safety engineering best practices as defined in international standards and guidelines [17][25][33][34][36][39] and other sources on safety engineering. The limited number of basic patterns currently available delimits what can be modelled in a composite pattern. Defining more basic patterns will provide a better coverage of the tacit knowledge that can be externalised. A user can easily extend the language. A basic pattern, e.g., the pattern *Hazard Analysis* [6] referenced in Fig. 3, is defined in a simple structure of named sections containing text and illustrations according to a common format. The format is thoroughly detailed in [10].

Table II compares the overall format of basic SaCS patterns to pattern formats in the literature. We have chosen a format that resembles that of Alexander et al. [3] with the addition of the sections “Pattern signature”, “Intent”, “Applicability”, and “Instantiation rule”. The signature, intent, and applicability sections of basic patterns are documented in such a manner that the context section provided in [3] is not needed. The format in [3] is a suitable basis as it is simple, well-known, and generally applicable for specifying patterns of different kinds. The format provided by Gamma et al. [13] is also simple and well-known, but tailored specifically for capturing patterns for software design.

All in all, we admit that there can be relevant tacit knowledge that is not easily externalised as the SaCS language is today. However, the opportunity of increasing the number of basic patterns makes it possible to at least reduce the gap.

MA.2 The language must support the modelling needs within conceptual safety design.

IEC 61508 [25] is defined to be applicable across all industrial domains developing safety-related systems. As already mentioned, a key concept within IEC 61508 is functional safety. Functional safety is achieved, according to [25], by adopting a broad range of principles, techniques and measures.

A key concept within SaCS is that principles, techniques, methods, activities, and technical solutions of different kinds are defined within the format of basic patterns. A limited number of concerns are addressed by each basic pattern. A specific combination of patterns is defined within a composite pattern. A composite pattern is intended to address the overall challenges that appear in a given development context. Individual patterns within a composite only address a subset of the challenges that need to be solved in the context. A composite can be defined prior to work initiation in order to define a plan for the application of patterns. A composite that is defined as a representation of a work plan can be easily reused for documentation purposes by adding information on the instantiation of parameters. Another use can be to refine a composite throughout the work process. This is exemplified in [6] and [7]. A composite can also be defined once patterns have been applied in order to document the work process.

C. Participants appropriateness

PA.1 The terms used for concepts in the language must be the same terms used within safety engineering.

Activities such as *hazard identification* and *hazard analysis* [34], methods such as *fault tree analysis* [27] and *failure mode effects analysis* [26], system design solutions including *redundant modules* and voting mechanisms [32], and practices like arguing safety on the basis of arguing that safety requirements are satisfied [36], are all well known safety engineering practices that can be found in different standards and guidelines [17][25][39]. The different concepts mentioned above are all reflected in basic SaCS patterns. Moreover, as already pointed out, keywords such as process assurance, product assurance, requirement, solution, safety case, etc. have all been selected based on leading terminology within safety engineering.

TABLE II. PATTERN FORMATS IN THE LITERATURE COMPARED TO BASIC SACS PATTERNS [10]

	[11]	[3]	[13]	[14]	[15]	[40]	[41]	[42]	[10]
Name	✓	✓	✓	✓	✓	✓	✓	✓	✓
Also known as			✓		✓				
Pattern signature									✓
Intent			✓		✓				✓
Motivation			✓		✓				
Applicability			✓		✓				✓
Purpose							✓		
Context	✓	✓				✓	✓		
Problem	✓	✓		✓		✓		✓	✓
Forces	✓					✓			
Solution	✓	✓		✓		✓	✓	✓	✓
Structure			✓		✓				
Participants			✓		✓				
Collaborations			✓		✓				
Consequences			✓		✓				
Implementation			✓		✓				
Sample code			✓						
Example					✓				
Compare							✓		
Instantiation rule									✓
Related patterns		✓	✓	✓	✓	✓			✓
Known uses	✓	✓		✓		✓			✓

PA.2 The symbols used to illustrate the meaning of concepts in the language must reflect these meanings.

One commonly cited and influential article within psychology is that of Miller [43], on the limit of human capacity to process information. The limit, according to Miller, is seven plus or minus two elements. When the number of elements increases past seven, the mind can be confused in correctly interpreting the information. Thus, the number of symbols should be kept low in order to facilitate effective human information processing.

Lidwell et al. [44] describe iconic representation as *"the use of pictorial images to make actions, objects, and concepts in a display easier to find, recognize, learn, and remember"*. The authors describe four forms for representation of information with icons: similar, example, symbolic, and arbitrary. We have primarily applied the symbolic form to identify a concept at a higher level of abstraction than what can be achieved with the similar and example forms. We have also tried to avoid the arbitrary form where there is little or no relationship between a concept and its associated icon. Fig. 4, Fig. 5, Fig. 6, Fig. 7, and Fig. 8 present the main icons in SaCS. In order to allow a flexible use of icons and keep the number of icons low, we have chosen not to define a dedicated icon for each concept but rather define icons that categorises several related concepts. A relatively small number of icons was designed in a uniform manner in order to capture intuitive representations of related concepts. As an example, the referenced basic patterns in Fig. 3 have the same icons linking them by category, but unique identifiers separating them by name.

PA.3 The language must be understandable for people familiar with safety engineering without specific training.

The SaCS language is simple in the sense that a small set of icons and symbols are used for modelling the application of patterns, basically: pattern references as in Fig. 5, parameters and artefact references as in Fig. 7, relations as in Fig. 6, and instantiation order as in Fig. 8. Guidance to the understanding of the language is provided in [10], where the syntax and the semantics of SaCS patterns are described in detail. The SaCS language comes with a structured semantics [10] that offers a schematic mapping from syntactical elements into text in English. Guidance to the application of SaCS is provided by the examples detailed in [6] and [7]. Although we have not tested SaCS on people unfamiliar with the language, we expect that users familiar with safety engineering can comprehend the concepts and the modelling on the basis of the descriptions in [6][7][10] within 2-3 working days.

D. Comprehensibility appropriateness

CA.1 The concepts and symbols of the language should differ to the extent they are different.

The purpose of the graphical notation is to represent a structure of patterns in a manner that is intuitive, comprehensible, and that allows efficient visual perception. According to Larkin and Simon [45], the key activities performed by a reader

in order to draw conclusions from a diagram are: searching; and recognising relevant information.

Lidwell et al. [44] present 125 patterns of good design based on theory and empirical research on visualisation. The patterns describe principles of designing visual information for effective human perception. The patterns are defined on the basis of extensive research on human cognitive processes. Some of the patterns are commonly known as Gestalt principles of perception. Ellis [46] provides an extensive overview of the Gestalt principles of perception building upon classic work from Wertheimer [47] and others. Gestalt principles capture the tendency of the human mind to naturally perceive whole objects on the basis of object groups and parts.

One of several Gestalt principles applied in the SaCS language is the principle of similarity. According to Lidwell et al. [44], the principle of similarity is such that similar elements are perceived to be more related than elements that are dissimilar.

The use of the similarity principle is illustrated by the composite pattern in Fig. 3. Although each referenced pattern has a unique name, their identical icons indicate relatedness. Different kinds of patterns are symbolised by the icons in Fig. 4. The icons are of the same size with some aspects of similarity and some aspects of dissimilarity such that a degree of relatedness can be perceived. An icon for pattern reference is different in shape and shading compared to an icon used for artefact reference, see Fig. 4 and Fig. 7, respectively. Thus, an artefact and a pattern should be perceived as representing quite different concepts.

CA.2 It must be possible to group related statements in the language in a natural manner.

There are five ways to organise information according to Lidwell et al. [44]: category, time, location, alphabet, and continuum. The *category* refers to the organisation of elements by similarity and relatedness. An example of the application of the principle of categorisation [44] in SaCS is seen in the possibility to reduce the number of relations drawn between patterns when these are similar. Patterns in SaCS can have multiple inputs and multiple outputs as indicated in Fig. 3. Relations between patterns operate on the parameters. The brackets [ ] placed adjacent to a pattern reference denotes an ordered list of parameters. In order to avoid drawing multiple relations between two patterns, relations operate on the ordered parameter lists of the patterns by list-matching of parameters.

Fig. 21 exemplifies two different ways for expressing visually the same relationships between the composite patterns named *A* and *B*. The list-matching mechanism is used to reduce the number of relation symbols drawn between patterns to one, even though the phenomena modelled represents multiple similar relations. This reduces the visual complexity and preserves the semantics of the relationships modelled.

CA.3 It must be possible to reduce model complexity with the language.

Hierarchical organisation is the simplest structure for visualising and understanding complexity according to Lidwell et al. [44]. The SaCS language allows concepts to be organised hierarchically by specifying that one pattern is detailed by another or by defining composite patterns that reference other composite patterns in the content part.

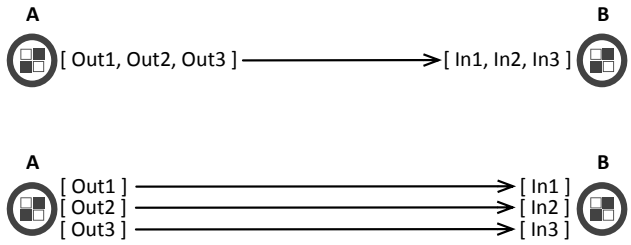


Figure 21. Alternative ways for visualising multiple similar relations

Fig. 22 presents a composite pattern named *Requirements* that reference other composites as part of its definition. The contained pattern *Safety Requirements* is defined in Fig. 3. The contained pattern *Functional Requirements* is not defined and is referenced within Fig. 22 for illustration purposes. *Requirements* can be easily extended by defining composites supporting the elicitation of, e.g., performance requirements and security requirements, and later model the use of such patterns in Fig. 22. In Fig. 22, the output of applying the *Requirements* pattern is represented by the parameter *ReqSpec*. The *ReqSpec* parameter represents the result of applying the *combines* relation on the output *Req* of the composite *Safety Requirements* and the output *Req* of the composite *Functional Requirements*.

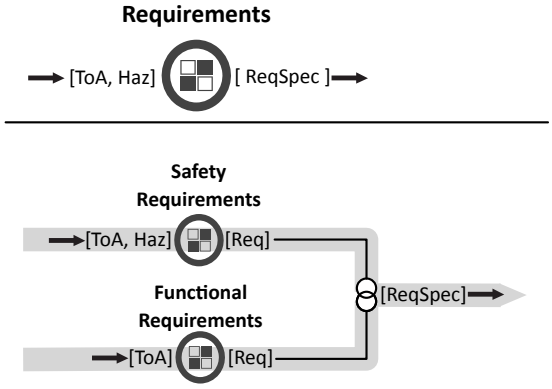


Figure 22. Composition of composites

CA.4 The symbols of the language should be as simple as possible with appropriate use of colour and emphasis.

A general principle within visualisation according to Lidwell et al. [44] is to use colour with care as it can lead to misconceptions if used inappropriately. The authors points out that there is no universal symbolism for different colours. As colour blindness is common the SaCS language applies different shades of grey in visualisations.

Moody [48] defines 9 principles for designing cognitive effective visual notations optimised for human understanding. The principles are synthesised from theory and empirical research from a wide range of fields. We firstly introduce the set of principles. Secondly, we exemplify and discuss the

implementation of the principles in the SaCS pattern language. The principles are:

1) *Semiotic clarity*: concerns to which extent there is a correspondence between semantic constructs and graphical symbols [48].
2) *Perceptual discriminability*: concerns to which extent different symbols are distinguishable [48].
3) *Semantic transparency*: concerns to which extent the meaning of a symbol can be inferred from its appearance [48].
4) *Complexity management*: concerns to which extent the visual notation is able to represent information without overloading the human mind [48].
5) *Cognitive integration*: concerns to which extent there are explicit mechanisms supporting the integration of information from different diagrams [48].
6) *Visual expressiveness*: concerns to which extent the full range of and capacities of visual variables are used [48].
7) *Dual coding*: concerns to which extent text is used to complement graphics [48].
8) *Graphic economy*: concerns to which extent the number of different graphical elements are cognitively manageable [48].
9) *Cognitive fit*: concerns to which extent visual dialects are used to support different tasks and audiences [48].

Fig. 23 presents how the principles 1-9 outlined above are implemented. Fig. 23 also presents how three of the Gestalt principles of visual perception [47][46][44] known as Figure-Ground, Proximity, and Uniform connectedness are implemented. The Gestalt principles express mechanisms for efficient human perception from groups of visual objects. In Fig. 23, the coloured elements represent the annotations used for explaining the implementation of principles, while the remaining elements are SaCS notation. Below we give a further description of the implementation of the 9 principles.

Regarding 1): We have deliberately applied a symbol deficit approach, which influences semiotic clarity [48]. Several concepts are expressed in natural language rather than by symbols, e.g., names of patterns and parameters. Hence, there is not a one-to-one mapping between concepts and symbols. We believe an approach where all concepts are symbolised with dedicated symbols is counter-productive. A one-to-one mapping between concepts and symbols would likely lead to symbol overload as well as violate the Gestalt principle of simplicity [44]. There is also a limit of human capacity to process information that motivates a small number of symbols. The limit, which was mentioned earlier, is according to Miller [43] seven plus or minus two elements. In order to facilitate effective human information processing we have defined a small set of symbols. The different icons are used as classifiers instead of being associated with one specific concept. However, any lack of clarity with our symbol deficit approach is compensated by the use of other principles such as dual coding explained later.

Regarding 2): We have approached perceptual discriminability [48] by using the Gestalt principle of similarity to balance the need for similarity with the need for dissimilarity. The principle of similarity [44] is such that similar elements are perceived to be more related than elements that are dissimilar. The use of the similarity principle can be seen applied to the icons for the different contained patterns in Fig. 3. Although each referenced pattern has a unique name, their identical icons indicate relatedness. Different kinds of patterns are symbolised by the icons in Fig. 4. The icons are of the same size with some aspects of similarity and some aspects of dissimilarity such that a degree of relatedness may be perceived. Textual differentiation is approached with different typefaces and font size.

Regarding 3): We described earlier that our aim when designing icons was to achieve a symbolic form according to the classification of Lidwell et al. [44]. The effect is icons that identify a concept at a higher level of abstraction than can be achieved with the preferable similar and example forms, but at least the arbitrary form is avoided where there is little or no relationship between a concept and its associated icon. Section II-C presents the main icons in SaCS and identifies the result of our efforts to achieve semantic transparency [48].

Regarding 4): Several different mechanism are used for complexity management [48] in SaCS. According to Lidwell et al. [44], hierarchical organisation is the simplest structure for visualising and understanding complexity. The SaCS language offers different kinds of hierarchical organisation, e.g., the *details* relation may be used to specify that one pattern is detailed by another, and a composite pattern can use other composites as part of its specification. The latter provides a mechanism for modularisation.

Regarding 5): Different kinds of basic patterns are integrated as support for conceptual safety design in SaCS. We have applied UML [49], GSN [50], and Problem Frames [51] to support modelling different kinds of concepts within basic patterns. These three languages are widely known, and no single language exists that serves the different modelling needs these notations offer. In addition, a principle of good design [52][44][53] is to balance the need for performance by the importance of preference in designing solutions. However, the choice of several languages challenges cognitive integration [48]. Kim et. al [54] argue within their theory on cognitive integration of diagrams that in order for a multi-diagram representation to be cognitively effective, a mechanism that supports conceptual as well as perceptual integration must be explicitly included. The annotations added to the UML activity diagram presented in Fig. 2 represents the SaCS specific mechanism that allows cognitive integration. The annotations facilitates the user in mapping parameters with diagram elements and are briefly described in Section II-C, fully defined in [10].

Regarding 6): The degree of visual expressiveness is defined by the number of visual variables used in a language [48]. Bertin [55] identifies 8 visual variables divided into 2 planar variables and 6 retinal variables. The planar variables are horizontal and vertical position. The retinal variables are shape, size, brightness, orientation, texture, and colour. Every visual variable besides colour is used either to encode information or attract the visual attention of the user to what is important. A general principle within visualisation is to use colour with care

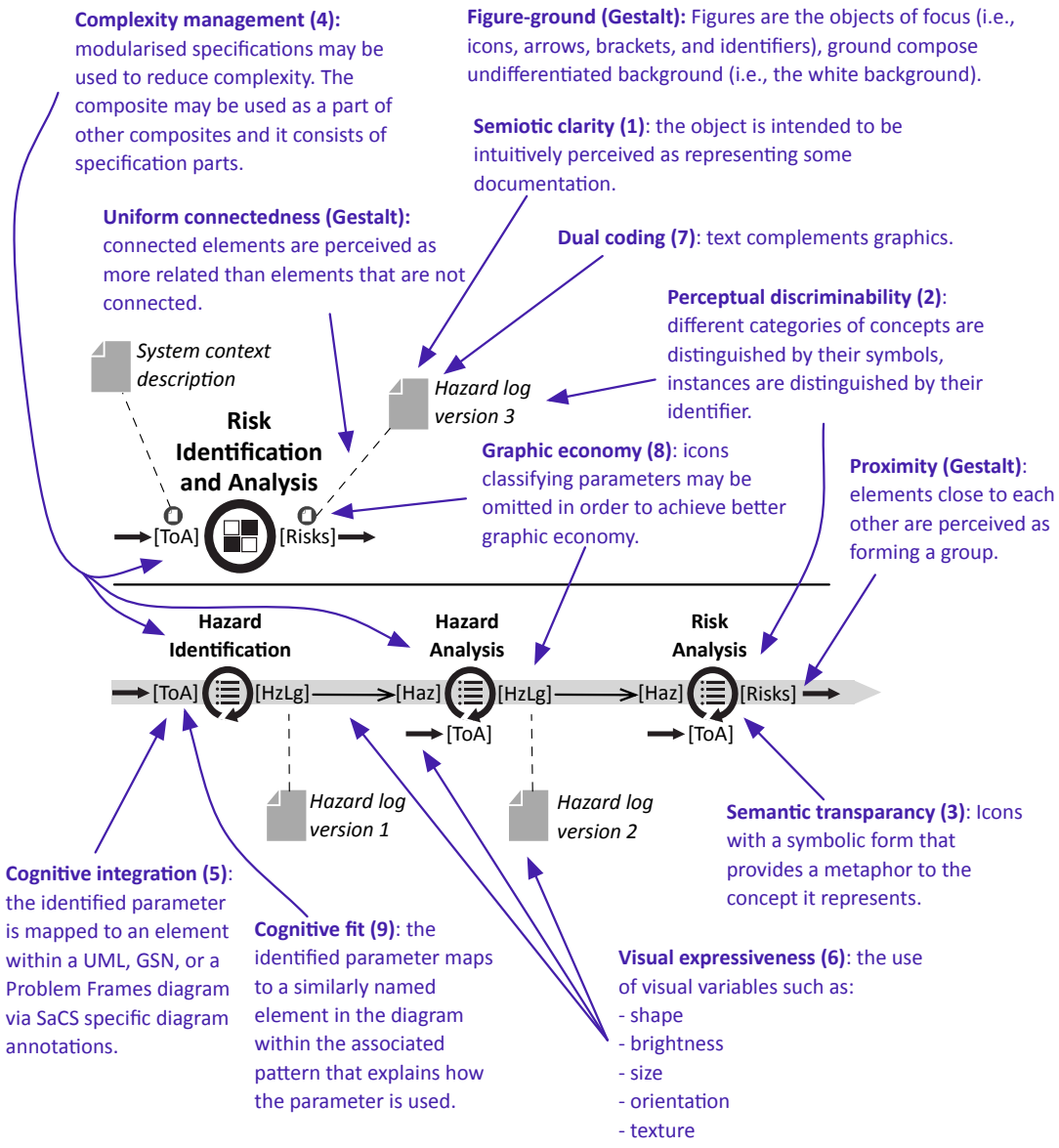


Figure 23. Example composite pattern with annotations showing the implementation of principles for cognitive effective visual notations

as it may lead to misconceptions if used inappropriately [44]. Moody [48] points out that although color is one of the most effective visual variables it should not be used as the sole basis for distinguishing between symbols, but rather for redundant coding. In this sense, colour coding can be added to our models for redundant coding and for emphasising particularly interesting information that requires immediate attention.

Regarding 7): Paivio [56] argues that within the dual coding theory, text and graphics together is a more effective carrier of information than using them separately. In SaCS, text is solely used for identifiers, e.g., identifiers for parameters, patterns, and development artefacts. Icons provide visual cues to what an entity represent. We believe this is a suitable strategy as identifiers are used in the verbal communication between users to name the entities that are discussed.

Regarding 8): As mentioned earlier we have deliberately applied a symbol deficit approach in composite patterns, which reduces the number of graphical symbols and positively effect graphic economy [48]. We have used the dual coding principle to balance text with symbols, where symbols classify entities and text provides supplementing information. Information is primarily textual in basic patterns. Basic patterns provide detailed guidance on different concepts that is difficult to fully capture graphically. Thus, diagrams are used in basic patterns to provide supplementing information.

Regarding 9): The intended users of SaCS represent different engineering disciplines and roles. According to the cognitive fit theory [48], different kinds of information representation should be used depending on task and audience. In order to achieve an overall effective visual representation, visual dialects suited to the individual tasks should be integrated rather than providing one representation for all purposes. A

requirements engineer is expected to be aware of the Problem Frames [51] notation. A system engineer, hardware engineer, or a software engineer is expected to be aware of the UML [49] notation. A safety engineer is expected to be aware of the the GSN [50] notation. The SaCS pattern language integrates these visual dialects and facilitates the communication between users on the development of conceptual safety design.

E. Tool appropriateness

TA.1 The language must have a precise syntax.

The syntax of the SaCS language (see [10]) is defined in the EBNF [57] notation. EBNF is a meta-syntax that is widely used for describing context-free grammars.

TA.2 The language must have a precise semantics.

A structured semantics for SaCS patterns is defined in [10] in the form of a schematic mapping from pattern definitions, via its textual syntax in EBNF [57], to English. The non-formal representation of the semantics supports human interpretation rather than tools, although the translation procedure as described in [10] can be automated. The presentation of the semantics of patterns as a text in English was chosen in order to aid communication between users, possibly with different technical background, on how to interpret patterns.

F. Organisational appropriateness

OA.1 The language must be able to express the desired conceptual safety design when applied in a safety context.

The application of the SaCS pattern language produces composite patterns that are instantiated into conceptual safety designs. A composite pattern expresses a combination of basic patterns. The basic patterns express safety engineering best practices and concepts inspired by international safety standards and guidelines, e.g., [17][25][39]. International safety standards and guidelines describe concepts and practices for development of safety critical systems that can be perceived as commonly accepted. The SaCS pattern language is tested out in two cases. The first concerned the conceptualisation of a nuclear power plant control system, while the second addressed the conceptualisation of a railway interlocking system, fully detailed in [6] and [7], respectively. In both cases it was possible to derive a conceptual safety design using the SaCS language as support as well as model how patterns were applied as support.

OA.2 The language must ease the comprehensibility of best practices within conceptual safety design for relevant target groups like system engineers, safety engineers, hardware and software engineers.

We have already explained how basic patterns represent concepts and best practices inspired by safety standards and guidelines. Each basic pattern addresses a limited number of phenomena. Basic patterns are combined into a composite pattern where the composite addresses all relevant challenges that occur in a specific context. A composite pattern as the one presented in Fig. 3 eases the explanation of how several concepts within conceptual safety design are combined and applied.

Wong et al. [58] reviewed several large development projects and software safety standards from different domains with respect to cost-effectiveness. Their conclusion is that although standards provide useful and effective guidance, safety and cost-effectiveness objectives are met by effective planning and by applying safety engineering best practices evidenced in company best practices throughout the development life cycle. A composite pattern can be easily defined with the SaCS pattern language in order to capture a company specific practice. In order to capture different company practices, different compositions of patterns can be defined.

OA.3 The language must be usable without the need of costly tools.

Every pattern used in the cases described in [6] and [7] was interpreted and applied in its context by a single researcher with background from safety engineering. A conceptual safety design was produced for each case. Every illustration in [6][7][10] and in this paper is created with a standard drawing tool.

VI. Related Work

In the literature, pattern approaches supporting development of safety critical systems are poorly represented. In the following we briefly discuss some different pattern approaches and their relevancy to the development of conceptual safety designs.

Jackson [51] presents the problem frames approach for requirements analysis and elicitation. Although the problem frames approach is useful for detailing and analysing a problem and thereby detailing requirements, the problem classes presented in [51] are defined on a very high level of abstraction.

The use of boilerplates [59][60] for requirement specification is a form of requirement templates but nonetheless touches upon the concept of patterns. The boilerplate approach helps the user phrase requirements in a uniform manner and to detail these sufficiently. Although boilerplates can be useful for requirement specification, the focus in SaCS is more towards supporting requirement elicitation and the understanding of the challenges that appear in a specific context.

Withall [61] describes 37 requirements patterns for assisting the specification of different types of requirements. The patterns are defined at a low level, i.e., the level of a single requirement. The patterns of Withall can be useful, but as with the boilerplates approach, the patterns support more the specification of requirements rather than requirements elicitation.

Patterns on design and architecture of software-based systems are presented in several pattern collections. One of the well-known pattern collections is the one of Gamma et al. [13] on recurring patterns in design of software based systems. Without doubt, the different pattern collections and languages on system design and architecture represent deep insight into effective solutions. However, design choices should be founded on requirements, and otherwise follow well established principles of good design. The choice of applying one design pattern over another should be based on a systematic process of establishing the need in order to avoid design choices being left unmotivated.

The motivations for a specific design choice are founded on the knowledge gained during the development activities applied prior to system design. Gnatz et al. [62] outline the concept of process patterns as a means to address the recurring problems and known solutions to challenges arising during the development process. The patterns of Gnatz et al. are not tailored for development of safety critical systems and thus do not necessarily reflect relevant safety practices. Fowler presents [12] a catalogue of 63 analysis patterns. The patterns do not follow a strict format but represent a body of knowledge on analysis described textually and by supplementary sketches.

While process patterns and analysis patterns can be relevant for assuring that the development process applied is suitable and leads to well informed design choices, Kelly [15] defines patterns supporting safety demonstration in the form of reusable safety case patterns. The patterns expressed are representative for how we want to address the safety demonstration concern.

A challenge is to effectively combine and apply the knowledge on diverse topics captured in different pattern collections and languages. Henninger and Corrêa [63] survey different software pattern practices and states *"software patterns and collections tend to be written to solve specific problems with little to no regard about how the pattern could or should be used with other patterns"*.

Zimmer [64] identifies the need to define relationships between system design patterns in order to efficiently combine them. Noble [65] builds upon the ideas of Zimmer and defines a number of relationships such as *uses, refines, used by, combine*, and *sequence of* as a means to define relationships between system design patterns. A challenge with the relations defined by Noble is that they only specify relations on a very high level. The relations do not have the expressiveness for detailing what part of a pattern is *used, refined*, or *combined*. Thus, the approach does not facilitate a precise modelling of relationships.

Bayley and Zhu [66] define a formal language for pattern composition. They argue that design patterns are almost always to be found composed with each other and that the correct applications of patterns thus relies on precise definition of the compositions. A set of six operators is defined for the purpose of defining pattern compositions. The language is exemplified on the formalisation of the relationships expressed between software design patterns described by Gamma et al. [13]. As we want the patterns expressed in the SaCS language to be understandable to a large community of potential users, we find this approach a bit too rigid.

Smith [67] presents a catalogue of elementary software design patterns in the tradition of Gamma et al. [13] and proposes the Pattern Instance Notation (PIN) for expressing compositions of patterns graphically. The notation uses simple rounded rectangles for abstractly representing a pattern and its associated roles. Connectors define the relationships between patterns. The connectors operate on the defined roles of patterns.

The notation of Smith [67] is comparable to the UML collaboration notation [49]. The main purpose of a UML collaboration is to express how a system of communicating entities collectively accomplishes a task. The notation is particularly suitable for expressing system design patterns.

Several notations [68][69][70] for expressing patterns graphically use UML as its basis. The notations are simple, but target the specification of software.

VII. Conclusion

We have presented an analytical evaluation of the SaCS pattern language with respect to six different appropriateness factors. We arrived at the following conclusions:

- *Domain*: In the design of the SaCS language we have as much as possible selected keywords and icons in the spirit of leading literature within the area. This indicates that we at least are able to represent a significant part of the concepts of relevance for conceptual safety design.
- *Modeller*: There can be relevant tacit knowledge that is not easily externalised as the SaCS language is today. However, the opportunity of increasing the number of basic patterns makes it possible to at least reduce the gap.
- *Participants*: The terms used for concepts have been carefully selected based on leading terminology within safety engineering. The SaCS language facilitates representing the application of best practices within safety design and mirror leading international standards; in particular IEC 61508.
- *Comprehensibility*: The comprehension of individual patterns and pattern compositions is supported by the use of terms commonly applied within the relevant industrial domains as well as by the application of principles of good design in visualisations, such as the Gestalt principles of perception [44][47].
- *Tool*: Tool support can be provided on the basis of the syntax and semantics of the SaCS language [10].
- *Organisational*: Organisations developing safety critical systems are assumed to follow a development process in accordance to what is required by standards. Wong et al. [58] reviewed several large development projects and software safety standards from different domains with respect to cost-effectiveness and concludes that although standards provide useful and effective guidance, safety and cost-effectiveness objectives are successfully met by effective planning and by applying safety engineering best practices evidenced in company best practices throughout the development life cycle. SaCS patterns can be defined, applied, and combined in a flexible manner to support company best practices and domain specific best practices.

Acknowledgment

This work has been conducted within the OECD Halden Reactor Project, Institute for Energy Technology, Halden, Norway. The second author has partly been funded by the ARTEMIS project CONCERTO.

REFERENCES

[1] A. A. Hauge and K. Stølen, "An Analytic Evaluation of the SaCS Pattern Language – Including Explanations of Major Design Choices," in Proceedings of the International Conference on Pervasive Patterns and Applications (PATTERNS'14). IARIA, 2014, pp. 79–88.

[2] F. Buschmann, K. Henney, and D. C. Schmidt, Pattern-Oriented Software Architecture: On Patterns and Pattern Languages. Wiley, 2007, vol. 5.

[3] C. Alexander, S. Ishikawa, and M. Silverstein, A Pattern Language: Towns, Buildings, Construction. Oxford University Press, 1977, vol. 2.

[4] J. C. Knight, "Safety Critical Systems: Challenges and Directions," in Proceedings of the 24th International Conference on Software Engineering (ICSE'02). ACM, 2002, pp. 547–550.

[5] J. E. McGrath, Groups: interaction and performance. Prentice-Hall, 1984.

[6] A. A. Hauge and K. Stølen, "A Pattern-based Method for Safe Control Conceptualisation – Exemplified Within Nuclear Power Production," Institute for Energy Technology, OECD Halden Reactor Project, Halden, Norway, Tech. Rep. HWR-1029 rev 2, 2014.

[7] A. A. Hauge and K. Stølen, "A Pattern-based Method for Safe Control Conceptualisation – Exemplified Within Railway Signalling," Institute for Energy Technology, OECD Halden Reactor Project, Halden, Norway, Tech. Rep. HWR-1037 rev 2, 2014.

[8] K. M. Eisenhardt, "Building theories from case study research," The Academy of Management Review, vol. 14, no. 4, 1989, pp. 532–550.

[9] J. Krogstie, Model-based Development and Evolution of Information Systems: A Quality Approach. Springer, 2012.

[10] A. A. Hauge and K. Stølen, "Syntax & Semantics of the SaCS Pattern Language," Institute for Energy Technology, OECD Halden Reactor Project, Halden, Norway, Tech. Rep. HWR-1052, 2013.

[11] A. Aguiar and G. David, "Patterns for Effectively Documenting Frameworks," in Transactions on Pattern Languages of Programming II, ser. LNCS, J. Noble, R. Johnson, P. Avgeriou, N. Harrison, and U. Zdun, Eds. Springer, 2011, vol. 6510, pp. 79–124.

[12] M. Fowler, Analysis Patterns: Reusable Object Models. Addison-Wesley, 1996.

[13] E. Gamma, R. Helm, R. E. Johnson, and J. Vlissides, Design Patterns: Elements of Reusable Object-Oriented Software. Addison-Wesley, 1995.

[14] R. S. Hanmer, Patterns for Fault Tolerant Software. Wiley, 2007.

[15] T. P. Kelly, "Arguing Safety – A Systematic Approach to Managing Safety Cases," Ph.D. dissertation, University of York, United Kingdom, 1998.

[16] B. Rubel, "Patterns for Generating a Layered Architecture," in Pattern Languages of Program Design, J. Coplien and D. Schmidt, Eds. Addison-Wesley, 1995, pp. 119–128.

[17] CENELEC, "EN 50129 Railway Applications – Communications, Signalling and Processing Systems – Safety Related Electronic Systems for Signalling," European Committee for Electrotechnical Standardization, 2003.

[18] J. Mendling, G. Neumann, and W. van der Aalst, "On the Correlation between Process Model Metrics and Errors," in Proceedings of 26th International Conference on Conceptual Modeling, vol. 83, 2007, pp. 173–178.

[19] A. G. Nysetvold and J. Krogstie, "Assessing Business Process Modeling Languages Using a Generic Quality Framework," in Proceedings of the 17th Conference on Advanced Information Systems Engineering (CAiSE'05) Workshops. Idea Group, 2005, pp. 545–556.

[20] J. Krogstie and S. D. F. Arnesen, "Assessing Enterprise Modeling Languages Using a Generic Quality Framework," in Information Modeling Methods and Methodologies. Idea Group, 2005, pp. 63–79.

[21] D. L. Moody, G. Sindre, T. Brasethvik, and A. Sølvberg, "Evaluating the Quality of Process Models: Empirical Testing of a Quality Framework," in Proceedings of the 21st International Conference on Conceptual Modeling, ser. LNCS. Springer, 2013, vol. 2503, pp. 380–396.

[22] J. Becker, M. Rosemann, and C. von Uthmann, "Guidelines of Business Process Modeling," in Business Process Management, ser. LNCS, vol. 1806. Springer, 2000, pp. 30–49.

[23] I. Hogganvik, "A Graphical Approach to Security Risk Analysis," Ph.D. dissertation, Faculty of Mathematics and Natural Sciences, University of Oslo, 2007.

[24] CENELEC, "EN 50126 Railway Applications - The Specification and Demonstration of Reliability, Availability, Maintainability and Safety (RAMS)," European Committee for Electrotechnical Standardization, 1999.

[25] IEC, "IEC 61508 Functional Safety of Electrical/Electronic/Programmble Electronic Safety-related Systems, 2nd Edition," International Electrotechnical Commission, 2010.

[26] IEC, "IEC 60812 Analysis Techniques for System Reliability – Procedure for Failure Mode and Effects Analysis (FMEA), 2nd edition," International Electrotechnical Commission, 2006.

[27] IEC, "IEC 61025 Fault Tree Analysis (FTA), 2nd edition," International Electrotechnical Commission, 2006.

[28] IEC, "IEC 61226 Nuclear Power Plants – Instrumentation and Control Important to Safety – Classification of Instrumentation and Control Functions, 3rd Edition," International Electrotechnical Commission, 2009.

[29] CENELEC, "EN 50128 Railway Applications – Communications, Signalling and Processing Systems – Software for Railway Control and Protection Systems," European Committee for Electrotechnical Standardization, 2001.

[30] Jernbaneverket, "Teknisk Regelverk, JD550 Prosjektering," https://trv.jbv.no/wiki/Signal/Prosjektering, 2014, [accessed: 2014-08-31].

[31] L. Sha, "Using Simplicity to Control Complexity," IEEE Software, vol. 18, 2001, pp. 20–28.

[32] N. Storey, Safety-critical Computer Systems. Prentice Hall, 1996.

[33] European Commission, "Commission Regulation (EC) No 352/2009 on the Adoption of Common Safety Method on Risk Evaluation and Assessment," Official Journal of the European Union, 2009.

[34] ERA, "Guide for the Application of the Commission Regulation on the Adoption of a Common Safety Method on Risk Evaluation and Assessment as Referred to in Article 6(3)(a) of the Railway Safety Directive," European Railway Agency, 2009.

[35] IEC, "IEC 60880 Nuclear Power Plants – Instrumentation and Control Systems Important to Safety – Software Aspects for Computer-based Systems Performing Category A Functions, 2nd Edition," International Electrotechnical Commission, 2006.

[36] The Members of the Task Force on Safety Critical Software, "Licensing of safety critical software for nuclear reactors: Common position of seven european nuclear regulators and authorised technical support organisations," http://www.belv.be/, 2013, [accessed: 2014-08-31].

[37] I. Habli and T. Kelly, "Process and Product Certification Arguments – Getting the Balance Right," SIGBED Review, vol. 3, no. 4, 2006, pp. 1–8.

[38] C. Haddon-Cave, "The Nimrod Review: An Independent Review Into the Broader Issues Surrounding the Loss of the RAF Nimrod MR2 Aircraft XV230 in Afghanistan in 2006," The Stationery Office (TSO), Tech. Rep. 1025 2008-09, 2009.

[39] IEC, "IEC 61513 Nuclear Power Plants – Instrumentation and Control Systems Important to Safety – General Requirements for Systems, 2nd Edition," International Electrotechnical Commission, 2001.

[40] A. Ratzka, "User Interface Patterns for Multimodal Interaction," in Transactions on Pattern Languages of Programming III, ser. LNCS, J. Noble, R. Johnson, U. Zdun, and E. Wallingford, Eds. Springer, 2013, vol. 7840, pp. 111–167.

[41] D. Riehle and H. Züllinghoven, "A Pattern Language for Tool Construction and Integration Based on the Tools and Materials Metaphor," in Pattern Languages of Program Design, J. Coplien and D. Schmidt, Eds. Addison-Wesley, 1995, pp. 9–42.

[42] K. Wolf and C. Liu, "New Client with Old Servers: A Pattern Language for Client/Server Frameworks," in Pattern Languages of Program Design, J. Coplien and D. Schmidt, Eds. Addison-Wesley, 1995, pp. 51–64.

[43] G. A. Miller, "The Magical Number Seven, Plus or Minus Two: Some Limits on Our Capacity for Processing Information," Psychological Review, vol. 63, no. 2, 1956, pp. 81–97.

[44] W. Lidwell, K. Holden, and J. Butler, Universal Principles of Design, 2nd ed. Rockport Publishers, 2010.

[45] J. H. Larkin and H. A. Simon, "Why a Diagram is (Sometimes) Worth Ten Thousand Words," Cognitive Science, vol. 11, no. 1, 1987, pp. 65–100.

[46] W. D. Ellis, A Source Book of Gestalt Psychology. The Gestalt Journal Press, 1997.

[47] M. Wertheimer, "Laws of Organization in Perceptual Forms," in A sourcebook of Gestalt Psychology, W. D. Ellis, Ed. Routledge and Kegan Paul, 1938, pp. 71–88.

[48] D. L. Moody, "The "Physics" of Notations: Towards a Scientific Basis for Constructing Visual Notations in Software Engineering," IEEE Transactions on Software Engineering, vol. 35, no. 6, 2009, pp. 756–779.

[49] OMG, "Unified Modeling Language Specification, Version 2.4.1," Object Management Group, 2012.

[50] J. Spriggs, GSN – The Goal Structuring Notation: A Structured Approach to Presenting Arguments. Springer, 2012.

[51] M. Jackson, Problem Frames: Analysing and Structuring Software Development Problems. Addison-Wesley, 2001.

[52] R. W. Bailey, "Performance versus Preference," in Proceedings of 37th Annual Meeting of the Human Factors and Ergonomics Society, vol. 37, 1993, pp. 282–286.

[53] J. Nilsen and J. Levy, "Measuring Usability: Performance vs. Preference," Communications of the ACM, vol. 37, no. 4, 1994, pp. 66–75.

[54] J. Kim, J. Hahn, and H. Hahn, "How Do We Understand a System with (So) Many Diagrams? Cognitive Integration Processes in Diagrammatic Reasoning," Information Systems Research, vol. 11, no. 3, 2000, pp. 284–303.

[55] J. Bertin, Semiology of Graphics: Diagrams, Networks, Maps. University of Wisconsin Press, 1983.

[56] A. Paivio, Mental Representations: A Dual Coding Approach. Oxford University Press, 1986.

[57] ISO/IEC, "14977:1996(E) Information Technology - Syntactic Metalanguage - Extended BNF," International Organization for Standardization / International Electrotechnical Commission, 1996.

[58] W. E. Wong, A. Demel, V. Debroy, and M. F. Siok, "Safe Software: Does It Cost More to Develop?" in Fifth International Conference on Secure Software Integration and Reliability Improvement (SSIRI'11), 2011, pp. 198–207.

[59] E. Hull, K. Jackson, and J. Dick, Requirements Engineering, 3rd ed. Springer, 2010.

[60] G. Sindre, "Boilerplates for Application Interoperability Requirements," in Proceedings of 19th Norsk konferanse for organisasjoners bruk av IT (NOKOBIT'12). Tapir, 2012.

[61] S. Withall, Software Requirement Patterns (Best Practices), 1st ed. Microsoft Press, 2007.

[62] M. Gnatz, F. Marschall, G. Popp, A. Rausch, and W. Schwerin, "Towards a Living Software Development Process based on Process Patterns," in Proceedings of the 8th European Workshop on Software Process Technology (EWSPT'01), ser. LNCS, vol. 2077. Springer, 2001, pp. 182–202.

[63] S. Henninger and V. Corrêa, "Software Pattern Communities: Current Practices and Challenges," in Proceedings of the 14th Conference on Pattern Languages of Programs (PLOP'07). ACM, 2007, pp. 14:1–14:19, article No. 14.

[64] W. Zimmer, "Relationships Between Design Patterns," in Pattern Languages of Program Design. Addison-Wesley, 1994, pp. 345–364.

[65] J. Noble, "Classifying Relationships Between Object-Oriented Design Patterns," in Proceedings of Australian Software Engineering Conference (ASWEC'98), 1998, pp. 98–107.

[66] I. Bayley and H. Zhu, "A Formal Language for the Expression of Pattern Compositions," International Journal on Advances in Software, vol. 4, no. 3, 2012, pp. 354–366.

[67] J. M. Smith, Elemental Design Patterns. Addison-Wesley, 2012.

[68] H. Byelas and A. Telea, "Visualization of Areas of Interest in Software Architecture Diagrams," in Proceedings of the 2006 ACM Symposium on Software Visualization (SoftVis'06), 2006, pp. 105–114.

[69] J. Dong, S. Yang, and K. Zhang, "Visualizing Design Patterns in Their Applications and Compositions," IEEE Transactions on Software Engineering, vol. 33, no. 7, 2007, pp. 433–453.

[70] J. M. Vlissides, "Notation, Notation, Notation," C++ Report, 1998, pp. 48–51.

Combining Genetic Algorithm and SMT into Hybrid Approaches to Web Service Composition Problem

Artur Niewiadomski
Institute of Computer Science
Siedlce University, Poland
e-mail: artur.niewiadomski@uph.edu.pl

Wojciech Penczek
Institute of Computer Science
PAN Warsaw and Siedlce University, Poland
e-mail: wpenczek@gmail.com

Jaroslaw Skaruz
Institute of Computer Science
Siedlce University, Poland
e-mail: jaroslaw.skaruz@uph.edu.pl

***Abstract*—The paper deals with the concrete planning problem – a stage of the Web Service Composition in the PlanICS framework, which consists in choosing the best service offers in order to satisfy the user query and to maximize the quality function. We introduce a novel planning technique based on a combination of a Genetic Algorithm (GA) with a Satisfiability Modulo Theories (SMT) solver, which allows to obtain better results than each of the methods separately. We give three versions of a hybrid algorithm. Two of them involve a modification of the standard GA in such a way that after every couple of iterations of GA, several top-ranked individuals are processed by the SMT-based algorithm in order to improve them. The third one exploits an SMT-solver in order to generate the initial populations for GA, which results in a substantial improvement in the overall algorithm efficiency. The paper presents experimental results, which seem to be very encouraging.**

***Keywords*-Web Service Composition; Concrete Planning; Genetic Algorithm; Satisfiability Modulo Theories; Hybrid Algorithm**

I. INTRODUCTION

This paper is an improved and extended version of the Service Computation 2014 conference paper "Genetic Algorithm to the Power of SMT: a Hybrid Approach to Web Service Composition Problem" [1]. This work introduces the two new versions of our hybrid algorithm, namely Semi-Random and InitPop Hybrid. The former is a slightly modified variant of the algorithm provided in the original paper, but due to the introduced changes it is also much more powerful. The latter implements our brand new concept of combining Satisfiability Modulo Theories and Genetic Algorithms. This is our main original contribution since the InitPop Hybrid algorithm is presented here for the very first time. In comparison with [1], this paper extends also the experimental results section and widely discusses the related work.

One of the fundamental ideas of Service-Oriented Architecture (SOA) [2] is to compose simple functionalities, accessible via well-defined interfaces, in order to realize more sophisticated objectives. The problem of finding such a composition is hard and known as the Web Service Composition (WSC) problem [2][3][4].

PlanICS [5] is a framework aimed at WSC, easily adapting existing real-world services. The main assumption in PlanICS is that all the web services in the domain of interest as well as the objects that are processed by the services, can be strictly classified in a hierarchy of *classes*, organised in an *ontology*. Another key idea is to divide the planning into several stages. The first phase deals with *classes of services*, where each class represents a set of real-world services, while the other phases work in the space of *concrete services*. The first stage produces an *abstract plan* composed of service classes [6]. Next, offers are retrieved by the Offer Collector (OC), a module of PlanICS, and used in the concrete planning (CP). As a result of CP, a *concrete plan* is obtained, which is a sequence of offers satisfying predefined optimization criteria. Dividing the planning process into the two planning phases allows to dramatically reduce the number of web services to be considered and so the number of inquires for offers.

This paper deals with the Concrete Planning Problem (CPP), shown to be NP-hard [7]. Our previous works employ several techniques to solve it: a Genetic Algorithm (GA) [8], numeric optimization methods [9] as well as Satisfiability Modulo Theories (SMT) Solvers [7]. The results of the extensive experiments show that the proposed methods are complementary, but every single one suffers from some disadvantages. The main disadvantage of an SMT-based solution often demonstrates in a long computation time, which is not acceptable in the case of a real-world interactive planning tool. On the other hand, a GA-based approach is relatively fast, but it yields solutions, which could be far from optimum and are found with a low probability. Thus, our aim is to exploit the advantages of both methods by combining them into a hybrid algorithm. This methodology and its implementation is the main contribution of the paper. We present here three versions of a hybrid algorithm. Two of them involve a modification of the standard GA in such a way that after every couple of iterations of GA, several top-ranked individuals are processed by the SMT-based algorithm in order to improve them [10]. The third one exploits an SMT-solver in order to generate the initial populations for GA, which results in a substantial improvement of the algorithm efficiency. Such an approach is novel, whereas several other "pure" and hybrid approaches to CPP have been defined. We discuss them in the next section.

II. RELATED WORK

Over the last few years, the concrete planning problem has been extensively studied in the literature. G. Canfora et al. [11] use a simple GA to obtain a good quality concrete plan. As optimization criteria the authors choose features commonly

referred to as Quality of Service (QoS), like the response time of a web service, its cost, availability, and reliability. An individual of GA representing a concrete plan is encoded as a vector of integer values, where each value identifies an offer and each position of the vector corresponds to a service type. While a concrete plan has to satisfy a number of constraints, the concrete planning problem is transformed to the constrained optimisation problem. The authors define a penalty function, which decreases the fitness values of the individuals not satisfying some constraints. Unfortunately, experimental study concerns only 25 services and up to 25 offers for each service. However, our approach, where the user is free to define an objective function using any of the available attributes, seems to be much more flexible.

Y. Wu et al. [12] transform CPP to a multi-criteria optimization problem and exploit GA to find a concrete plan. However, the authors present the experiments on a relatively small search space that could not provide valuable conclusions. Another version of GA is presented in [13], where special genetic operators are applied in order to find a concrete plan satisfying user constraints. Moreover, the individuals are generated basing on a set of initially found concrete plans using greedy heuristics. This idea allows to evolve only feasible individuals within the population and the algorithm must only assure that genetic operators do not provide unfeasible potential solutions. Thus, the idea of generating the initial population is somewhat similar to the one applied in our InitPop Hybrid.

Solving CPP using GA with a new version of crossover and mutation operators is presented in [14]. An adaptive crossover and a mutation operator are designed to increase convergence to a local minimum. Moreover, the idea of tabu list, which comes from the Tabu Search algorithm is applied. The experiments show that an applictaion of the modified version of GA gives better results than using the standard one.

Besides many papers discussing an application of GA to solve CPP [15][16][17], there are also papers applying more advanced algorithms. The authors of the paper [18], where CPP is viewed as a multi-criteria optimization problem, use the NSGA-II algorithm to obtain a set of good quality plans. Again, the attributes like the cost, the execution time, the service availability and a reputation stand for the optimization criteria. Instead of defining one fitness function with weights for each feature, the authors propose four separate fitness functions, each for a single attribute. Each plan obtained is optimal according to one of the fitness functions defined in the algorithm. Unfortunately, this result has been obtained thanks to considering small state spaces only, generated by 3 types of services and 15 instances of each type.

Different approaches of the Artificial Immune Systems have also been applied to solve CPP. In [19] a modified version of the CLONALG algorithm is used to a single-phase planning. The algorithm finds simultaneously abstract and concrete plans. Another modification of CLONALG is presented in [20]. At the beginning of the algorithm antibodies representing potential solutions are generated randomly, but it is assured that all of them are feasible. Contrary to the other works, in this algorithm an individual is encoded in a binary way. Similarly to a number of other approaches the authors use QoS features as the optimisation criteria.

Hybrid algorithms for WSC are also known in the literature. In [21] a modified version of the Particle Swarm Optimization algorithm has been used as a method for solving CPP. In this algorithm two additional genetic operators, namely crossover and mutation, are applied to obtain better results than in the standard algorithm.

Another approach consists in a combination of two evolutionary algorithms, Tabu Search and GA [22]. Similarly to ours, the experiments have been performed using randomly generated benchmarks. The authors examine the performance of the hybrid algorithm in comparison with the "pure" GA and Tabu Search. Although, this hybrid method finds good quality concrete plans, our hybrid algorithm allows for dealing with much larger search spaces. A hybrid approach based on an application of GA and the ant algorithm was proposed in [23]. The problem has been transformed to searching for the shortest path in a graph. While the experimental results are better than these obtained using a simple GA, the number of service types and offers used are not sufficient to draw general conclusions about the efficiency of this approach.

Thus, after a thorough analysis of the literature we can state that the main novelty of our approach consists not only in combining GA with SMT for solving CPP, but also in providing experimental results for benchmarks of very large state spaces.

The rest of the paper is structured as follows. In Section III the PlanICS framework is introduced and CPP is defined. Section IV presents the main ideas of our hybrid approach as well as some technical solutions. The experimental results are presented and discussed in Section V. The paper ends with some conclusions.

III. PLANICS FRAMEWORK

In this section we give an overview of the PlanICS framework. First, we briefly sketch the main concepts and the consecutive planning phases in order to eventually focus on the concrete planning stage. Then, we give all the definitions necessary to formulate the concrete planning problem. The section ends with an example planning scenario.

A. Overview of PlanICS

An ontology contains a set of *classes* describing the types of the services as well as the types of the objects they process. A class consists of a unique name and a set of the attributes. By an *object* we mean an instance of a class. By a *state* of an object we mean a valuation of its attributes. A set of objects in a certain state is called a *world*. A key notion of PlanICS is that of a *service*. We assume that each service processes a set of objects, possibly changing values of their attributes, and produces a set of new (additional) objects. We say that a service s *transforms* a world w into another world w'. A *transformation sequence* $s_1 \dots s_n$ is a sequence of services for which there is a sequence of worlds $w_0 \dots w_n$ such that s_i transforms w_{i-1} into w_i, for each $1 \leq i \leq n$. The types of services available for planning are defined as elements of the branch of classes rooted at the *Service* concept. Each service type stands for a description of a set of real-world services of similar functionality.

The main goal of the system is to find a composition of services that satisfies a user query. The query interpretation results in two sets of worlds: the initial and the expected ones. Moreover, the query may include additional constraints, especially *quality constraints*, the sum of which is used to choose the best solution from all the potential solutions. Thus, the task of the system is to find such a set of services, which transform some initial world into a world matching some expected one in such a way that the value of the quality function is maximized. Figure 1 shows the general PlanICS architecture. The bold arrows correspond to computation of a plan, the thin arrows model the planner infrastructure, while the dotted arrows represent the user interactions.

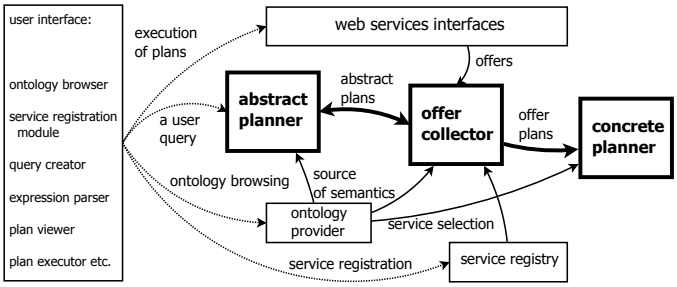


Figure 1. A diagram of the PlanICS system architecture.

The abstract planning is the first stage of the composition in the PlanICS framework. It consists in matching services at the level of input/output types and the abstract values. That is, since at this stage it is sufficient to know if an attribute does have a value or it does not, we abstract from the concrete values of the object attributes, and use two special values $\mathbf{set}$ and $\mathbf{null}$.

Thus, for a given ontology and a user query, the goal of the abstract planning is to find such a (multi)set of service types that allows to build a sequence of service types transforming an initial world of the user query into some *final world*. This final world has to be consistent with an expected world, which is also defined as a part of the query. The consistency between a final world and an expected one is expressed using the notion of the *compatibility* relation, formally defined in [6]. Intuitively, a final world W_f is compatible with an expected world W_e if the following conditions are satisfied:

- for every object $o_e \in W_e$ there exists a unique object $o_f \in W_f$, such that both the objects are of the same type or the type of o_f is a subtype of o_e,
- both the objects agree on the (abstract) values of the common attributes.

The result of the abstract planning stage is called a Context Abstract Plan (CAP). It consists of a multiset of service types (defined by a representative transformation sequence), contexts (mappings between the services and the objects being processed), and a set of final worlds. However, our aim is to find not only a single plan, but many (significantly different, and all if possible) abstract plans, in order to provide a number of alternative ways to satisfy the query. We distinguish between abstract plans built over different multisets of service types. See [6][24] for more details.

B. Concrete Planning Problem

In the second planning stage, a CAP is used by the Offer Collector, i.e., a tool, which in cooperation with the service registry, queries real-world services. The service registry keeps an evidence of real-world web services, registered accordingly to the service type system. During the registration, the service provider defines a mapping between the input/output data of the real-world service and the object attributes processed by the declared service type. OC communicates with the real-world services of types presented in a CAP. It sends the constraints on the data, which can potentially be sent to the service, and on the data expected to be received in an offer. Usually, each service type represents a set of real-world services. Moreover, querying a single service can result in a number of offers. Thus, we define offer sets as the main result of the second planning stage.

Definition 1 (Offer, Offer set): Assume that the n-th instance of a service type from a CAP processes some number of objects having in total m attributes. A single offer collected by OC is a vector $P = [v_1, v_2, \ldots, v_m]$, where v_j is a value of a single object attribute processed by n-th service of the CAP. An offer set O^n is a $k \times m$ matrix, where each row corresponds to a single offer and k is the number of offers in the set. Thus, the element $o^n_{i,j}$ is the j-th value of the i-th offer collected from the n-th service type instance from the CAP.

The responsibility of OC is to collect a number of offers, where every offer represents one possible execution of a single service. However, other important tasks of OC are: (1) building a set of constraints resulting from the user query and from semantic descriptions of service types, and (2) a conversion of the quality constraints expressed using objects from the user query to an *objective function* built over variables from offer sets. Thus, we can formulate CPP as a constrained optimization problem.

Definition 2 (CPP): Let n be the length of CAP and let $\mathbb{O} = (O^1, \ldots, O^n)$ be the vector of offer sets collected by OC such that for every $i = 1, \ldots, n$

$O^i = \begin{bmatrix} o^i_{1,1} & \cdots & o^i_{1,m_i} \\ \vdots & \ddots & \vdots \\ o^i_{k_i,1} & \cdots & o^i_{k_i,m_i} \end{bmatrix}$, and the j-th row of O^i is denoted by P^i_j. Let $\mathbb{P}$ denote the set of all possible sequences $(P^1_{j_1}, \ldots, P^n_{j_n})$, such that $j_i \in \{1, \ldots, k_i\}$ and $i \in \{1, \ldots, n\}$. The Concrete Planning Problem is defined as:

$$max\{Q(S) \mid S \in \mathbb{P}\} \text{ subject to } \mathbb{C}(S), \qquad (1)$$

where $Q : \mathbb{P} \mapsto \mathbb{R}$ is an objective function defined as the sum of all quality constraints and $\mathbb{C}(S) = \{C_j(S) \mid j = 1, \ldots, c$ for $c \in \mathbb{N}\}$, where $S \in \mathbb{P}$, is a set of constraints to be satisfied.

Finding a solution of CPP consists in selecting one offer from each offer set such that all constraints are satisfied and the value of the objective function is maximized. This is the goal of the third planning stage and the task of a concrete planner.

Example 1. Consider a simple ontology describing a fragment of some financial market consisting of service types inheriting

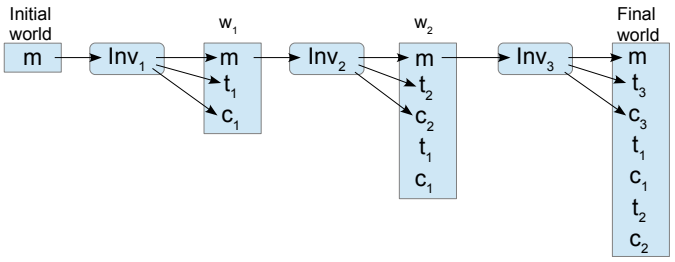


Figure 2. A graphical illustration of the context abstract plan given in the example.

from the class $Investment$, which represent various types of financial instruments. Moreover, the ontology contains three object types: $Money$ having the attribute $amount$, $Transaction$ having the two attributes $amount$ and $profit$, and $Charge$ having the attribute fee.

Suppose that each investment service takes m - an instance of $Money$ as input, produces t and c - instances of $Transaction$ and $Charge$, respectively, and updates the amount of money remaining after the operation, i.e., the attribute $m.amount$.

Assume that a user would like to invest up to \$100 in three financial instruments, locating more than \$50 in two investments. Moreover, the user wants to maximize the sum of profits and wants to use only services of handling fees less than \$3. The latter two conditions can be expressed as an appropriate quality function and an aggregate condition.

Formally, the user query can be formulated as follows: $in = \emptyset$, $inout$ = $\{m : Money\}$, out = $\{t_1, t_2, t_3 : \text{Transaction}\}$ pre= (m.amount $\leq$ 100), $post$ = (t_1.amount + t_2.amount $>$ 50), $Qual = \{(Transaction, true, profit, Sum)\}$, $Aggr = \{(Charge, true, fee, Max, <, 3)\}$.

The meaning of the consecutive components of the query is the following: in is a set of the read-only objects, while $inout$ is a set of the objects that could be modified - both are available at the start of the composition; out is a set of the objects that do not exist in the initial worlds, but they have to be produced by services of the plan. Next, pre and $post$ are boolean formulae describing the states of the initial and the expected worlds, respectively. Finally, $Qual$ and $Aggr$ are sets of quality constraints, and additional aggregation constraints, respectively.

Consider an exemplary CAP consisting of three instances of the $Investment$ service type depicted in Figure 2. The boxes represent worlds, the round boxes are services, while the arrows stand for transformation contexts.

A single offer collected by OC is a vector $[v_1, v_2, v_3, v_4, v_5]$, where v_1 corresponds to $m.amount$, v_2 to $t.amount$, v_3 to $t.profit$, and v_4 to $c.fee$. Since the attribute $m.amount$ is updated during the transformation, the offers should contain values from the world before and after the transformation. Thus v_5 stands for the value of $m.amount$ after modification.

Assuming that instances of $Investment$ return k_1, k_2, and k_3 offers in response to the subsequent inquiries, we obtain three offer sets: O^1, O^2, and O^3, where O^i is a $k_i \times 5$ matrix of offer values. The conditions from the query are translated to the following constraints: $C_1 := (o^1_{i_1,1} \leq 100)$ and $C_2 := (o^1_{i_1,2} + o^2_{i_2,2} > 50)$, where i_1, i_2, and i_3 are variables ranging over $1 \ldots k_i$. Moreover, the amount of money left after the operation is an input for the next investment. Thus, we have: $C_3 := (o^1_{i_1,5} = o^2_{i_2,1})$ and $C_4 := (o^2_{i_2,5} = o^3_{i_3,1})$. The aggregate condition is translated to the following constraint: $C_5 := \big(max(\{o^1_{i_1,4}, o^2_{i_2,4}, o^3_{i_3,4}\}) < 3\big)$, while the quality expression is translated to the quality constraint $Q := \sum_{j=1}^{3} o^j_{i_j,3}$.

IV. Hybrid Solutions

So far, we have made the following observations of the experiments with the "pure" SMT- and GA-based planners. The main disadvantage of the SMT-based solution is often a long computation time, which is not acceptable in the case of a real-world interactive planning tool. On the other hand, the GA-based approach is relatively fast, but it yields solutions that are far from optimum, and of low probability. Thus, our strategy is to delegate some sub-problems to be solved by an SMT-solver in such a way that the computation time is acceptable while the results allow to obtain better performance of GA. In order to evaluate the efficiency of our new hybrid algorithms, we use as benchmarks several CPP instances, which can hardly be solved by "pure" planners. By comparing the performance of several versions of the hybrid algorithm (with various parameter combinations) on the same examples, we can conclude whether the hybrid algorithm outperforms each of the "pure" methods separately.

In this section we present the main ideas behind three hybrid algorithms, named: Random Hybrid (RH) [1], Semi-Random Hybrid (SRH) [10], and InitPop Hybrid (IPH). We start with a description of their common features.

A. Overview

Our hybrid approach is based on the standard GA aimed at solving CPP. GA is a non deterministic algorithm maintaining a population of potential solutions during an evolutionary process. A potential solution is encoded in a form of a GA individual, which, in case of CPP, is a sequence of natural values. In each iteration of GA, a set of individuals is selected for applications of genetic operations, such as the standard one-point crossover and mutation, which leads to obtaining a new population passed to the next iteration of GA. The selection of an individual, and thus the promotion of its offspring to the next generation depends on the value of the *fitness function*. The fitness value of an individual is the sum of the optimization objective and the ratio of the number of the satisfied constraints to the number of all the constraints (see Definition 2), multiplied by some constant β:

$$fitness(I) = q(S_I) + \beta \cdot \frac{|sat(\mathbb{C}(S_I))|}{c}, \qquad (2)$$

where I stands for an individual, S_I is a sequence of the offer values corresponding to I, $sat\big(\mathbb{C}(S_I)\big)$ is a set of the constraints satisfied by a candidate solution represented by I, and c is the number of all constraints. The role of β is to reduce both the components of the sum to the same order of magnitude and to control the impact of the components on the

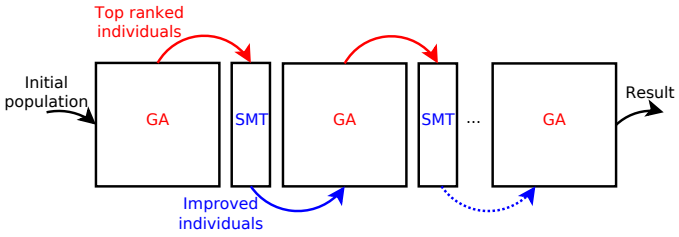


Figure 3. The RH and SRH algorithm overview.

final result. The value of β depends on the estimation of the minimal and the maximal quality function value.

B. Random and Semi-Random Hybrid Algorithms

The RH and SRH algorithms are based on the following modification of the standard GA (see Figure 3). After every couple of iterations of GA, several top-ranked individuals are processed by the SMT-based algorithm. Given an individual I, the procedure searches for a similar, but improved individual I', which represents a solution satisfying all the constraints and having a greater value of the objective function at the same time. The similarity between I and I' consists in sharing a number of genes. We refer to the problem of finding such an individual as to the *Search for an Improved Individual* (*SFII*). Since there are many possible ways to exploit this idea, we start from the one that randomly selects the genes to be changed.

The SMT procedure combined with GA is based on the encoding exploited in our "pure" SMT-based concrete planner [7][9]. The idea is to encode *SFII* as an SMT formula that is satisfiable if such an individual exists. First, we initialize an SMT-solver allocating a set $\mathcal{V}$ of all necessary variables:

- $\mathbf{oid}^i$, where $i = 1 \ldots n$ and n is the length of the abstract plan. These variables are needed to store the identifiers of offers constituting a solution. A single $\mathbf{oid}^i$ variable takes a value between 1 and k_i.
- $\mathbf{o}^i_j$, where $i = 1 \ldots n$, $j = 1 \ldots m_i$, and m_i is the number of offer values in the i-th offer set. We use them to encode the values of S, i.e., the values from the offers chosen as a solution. From each offer set O^i we extract the subset R^i of offer values that are present in the constraint set and in the quality function, and we allocate only the variables relevant for the plan.

Next, using the variables from $\mathcal{V}$, we encode the offer values, the objective function, and the constraints, as the formulas shared by all calls of our SMT-procedure. The offer values from the offer sets $\mathbb{O} = (O^1, \ldots, O^n)$ are encoded as the formula

$$ofr(\mathbb{O}, \mathcal{V}) = \bigwedge_{i=1}^{n} \bigvee_{d=1}^{k_i} \Big(\mathbf{oid}^i = d \ \wedge \bigwedge_{o^i_{d,j} \in R^i} \mathbf{o}^i_j = o^i_{d,j} \Big). \quad (3)$$

The formulae $ctr\big(\mathbb{C}(S), \mathcal{V}\big)$ and $qual\big(Q(S), \mathcal{V}\big)$, denoted as $\mathbf{ctr}$ and $\mathbf{q}$ for short, encode the constraints and the objective function, respectively. Details are provided in [7].

Let $I = (g_1, \ldots, g_n)$ be an individual, $M = \{i_1, \ldots, i_k\}$ the set of indices of genes allowed to be changed, and $q(S_I)$ the value of the objective function where $n, k \in \mathbb{N}$.

Hence, the *SFII* problem is reduced to the problem of satisfiability of the following formula:

$$\bigwedge_{i \in \{1,\ldots,n\} \setminus M} (\mathbf{oid}^i = g_i) \wedge ofr\big(\mathbb{O}, \mathcal{V}\big) \wedge \mathbf{ctr} \wedge (\mathbf{q} > q(S_I)) \quad (4)$$

That is, the Formula (4) is satisfiable only if there exists an individual $I' = (g'_1, \ldots, g'_n)$ satisfying all the constraints, where $\forall_{i \notin M}\ g_i = g'_i$ and $q(S_{I'}) > q(S_I)$, i.e., sharing with I all genes of indices outside M and having the larger value of objective function than I. If the formula is satisfiable, then the values of the changed genes are decoded from the model returned by the SMT-solver, and the improved individual I' replaces I in the current population.

Although, we have presented the general idea of a hybrid algorithm, there are still a number of problems that need to be solved in order to combine GA and SMT. Moreover, they can be solved in many different ways. The crucial questions that need to be answered are as follows. When to start the *SFII* procedure for the first time? How many times and how often *SFII* should be run? How many genes should remain fixed? How to choose genes to be changed? Since there are many possibilities to deal with the above problems, we started from the simplest solution that randomly selects genes to be changed. The solutions to the remaining questions we treat as parameters in order to develop the first version of our hybrid solution, called Random Hybrid (RH). Its pseudo-code is presented in Algorithm 1.

After analysing the experimental results (see Section V) we have found that the results are slightly better than these obtained using GA and SMT separately, however they could still be improved, especially in terms of a higher probability and a lower computation time. Thus, we introduced several improvements to the RH algorithm and we implemented the Semi-Random Hybrid (SRH) algorithm. The most important improvements introduced to SRH are as follows.

The $selectGenes$ procedure (see the line 11 of Algorithm 1) is not completely random any more. In the first place the genes violating some constraints are chosen to be changed. Then, the additional gene indices are selected randomly until we get a set of size gn. This change allows to increase the probability of finding a solution.

The next improvement aims at reducing the computation time. It consists in running the *SFII* procedure only if an individual violates some constraints. Thus, in case of SRH the lines from 11 to 15 in Algorithm 1 are executed conditionally, only if the individual I violates some constraints. In Section V we discuss the results obtained and compare both the approaches with the third hybrid solution - the IPH algorithm.

C. The InitPop Hybrid Algorithm (IPH)

The third hybrid algorithm also combines GA with SMT, but it does it in a different way. Our observations of the behaviour of the RH and SRH algorithms have had a big influence on the third hybrid algorithm. The first conclusion is

```
1  Procedure RandomHybrid(st, ind, int, gn, N)
   Input: st: when to start SFII for the first time,
   ind: the number of individuals to pass to SFII during a single
   GA iteration,
   int: how often run SFII,
   gn: the number of genes to be changed by SFII,
   N: the number of GA iterations.
   Result: an individual representing the best concrete plan
           found, or null
2  begin
3      initialize(); // generate initial
       population, initialize SMT solver
4      evaluate(); // compute fitness function for
       all individuals
5      for (i ← 1; i ≤ N; i ← i + 1) do
6          selection(); crossover(); mutation();
           // ordinary GA routines
7          evaluate();
8          if (i ≥ st) ∧ (i mod int = 0) then
9              BI ← findBestInd(ind); // a set of
               ind top individuals
10             foreach I ∈ BI do
11                 M ← selectGenes(I, gn); // a set of
                   gene indices to be changed
12                 I′ ← runSFII(M);
13                 if I′ ≠ null then
14                     I ← I′; // replace I by I′ in
                       the current population
15                 end
16             end
17         end
18     end
19     {best} ← findBestInd(1);
20     if constraintsSatisfied(best) then
21         return best; // if a valid solution has
           been found
22     else
23         return null
24     end
25 end
```

Algorithm 1: A pseudocode of the RandomHybrid algorithm

that the larger number of constraints, the worse performance of GA. However, a large number of constraints does not worsen the efficiency of the SMT-based components. On the other hand, searching for individuals of quality higher than a given value is quite expensive for an SMT-solver, while this is a natural application of GA. Therefore, the main idea is to divide the tasks of both the modules so that each of them is doing its best.

Now, the SMT-solver is used to generate (a part of) the initial population for GA (see Figure 4). The individuals generated by the SMT-solver satisfy all constraints. Note that each such individual is already a solution of CPP, but typically its fitness value is not optimal. However, the individuals generated by an SMT-solver provide an "easy start" for the genetic algorithm. Moreover, since the initial population already contains at least one solution, the algorithm should *always* return a plan. On the other hand, if the collected offers do not allow to build a solution (or the constraints are contradictory), then we get a straight negative answer from the SMT-solver.

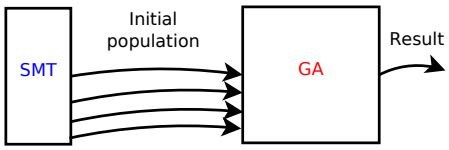


Figure 4. The IPH algorithm overview.

We have chosen a simple, but fast, strategy of generating individuals. That is, an individual is a valuation of the $\mathbf{oid}^1$, ..., $\mathbf{oid}^n$ variables satisfying the conjunction of the formula encoding the offers and the formula encoding the constraints: $ofr(\mathbb{O}, \mathcal{V}) \wedge \mathbf{ctr}$. Thus, in this case the sub-problem solved by the SMT-solver is much simpler than in the RH and SRH algorithms. In order to generate more than one individual, we also construct a formula blocking all valuations previously found, and this formula is conjuncted with the formula passed to the SMT-solver. That is, in order to generate the i-th individual the SMT-solver has to check the satisfiability of the following formula:

$$\varphi_i = ofr(\mathbb{O}, \mathcal{V}) \wedge \mathbf{ctr} \wedge \mathbf{B}_{i-1}. \tag{5}$$

Here, $\mathbf{B}_i$ stands for a blocking formula, which is defined inductively as:

$$\begin{aligned} \mathbf{B}_0 &= true, \\ \mathbf{B}_{i+1} &= \mathbf{B}_{i-1} \wedge \neg\Big(\bigwedge_{j=1..n} \mathbf{oid}^j = val_i(\mathbf{oid}^j) \Big), \end{aligned} \tag{6}$$

where $val_i(\mathbf{oid}^j)$ is the value of the j-th gene returned by the solver in the i-th step.

In the next section we provide the results of several experiments and compare the results of all three hybrid algorithms.

V. Experimental Results

In our previous experiments, discussed in [9], we have applied several "pure" methods to the concrete planning. These methods include the SMT-based approach, the standard GA, and numeric algorithms implemented in the OpenOpt toolset. The results of our experiments can be summarised as follows. The SMT-based planner is always able to find the optimal solution, provided that it is enough time and memory. In contrast, GA can find sometimes the optimal solution of length at most 5, but it consumes less time and memory. The ability of GA to find a concrete plan depends on the number of constraints. The more optimization constraints the smaller probability of finding a concrete plan. These drawbacks of GA are not common to our SMT-based approach. Moreover, our experiments show that a large number of constraints helps the SMT-solver to reduce the search space and to find the optimal solution faster. Overall, both the approaches are complementary and behave differently depending upon a particular problem instance. Concerning OpenOpt, we have shown that it can also be used for solving CPP. However, its effectiveness is satisfactory only if no tuples of values are present in the problem domain and much worse in the opposite case.

In order to evaluate the efficiency of our hybrid algorithms on "difficult" benchmarks, we have used for the experiments six instances of CPP that have been hardly solved with our "pure" SMT- and GA-based planners [7]. All the instances represent plans of length 15. Each offer set of Instance I,

TABLE I. EXPERIMENTAL RESULTS FOR INSTANCES I AND II.

				INSTANCE I						INSTANCE II					
				Random			Semi-Random			Random			Semi-Random		
exp	gn	ind	int	t[s]	Q	P	t[s]	Q	P	t[s]	Q	P	t[s]	Q	P
1	8	1	10	9.3	1305.0	3.3	9.3	1271.2	16.7	14.9	1382.0	6.7	15.9	1323.1	26.7
2			20	8.2	1331.5	6.7	7.9	1303.2	30.0	13.2	1371.5	13.3	12.3	1386.3	20.0
3		10	10	41.0	1386.7	53.3	25.7	1349.3	73.3	59.5	1437.6	36.7	40.0	1367.9	63.3
4			20	22.4	1389.0	26.7	17.9	1313.4	46.7	41.7	1414.0	33.3	31.4	1375.4	60.0
5		20	10	76.3	1405.8	70.0	36.4	1351.2	86.7	118.1	1441.0	73.3	66.3	1390.6	83.3
6			20	34.3	1356.5	43.3	24.9	1325.7	73.3	61.9	1420.3	40.0	43.5	1396.3	70.0
7	12	1	10	39.6	1363.1	66.7	**19.5**	**1337.7**	**86.7**	**56.6**	**1405.3**	**93.3**	30.0	1387.5	80.0
8			20	**14.5**	**1326.9**	**46.7**	11.0	1332.5	43.3	20.4	1380.0	40.0	**17.4**	**1369.9**	**73.3**
9		10	10	203.6	1417.6	100	74.8	1373.1	100	273.2	1455.8	100	108.1	1411.3	100
10			20	114.7	1362.2	100	54.0	1356.8	100	155.9	1431.3	100	76.5	1405.9	100
11		20	10	346.5	1424.2	100	122	1383.9	100	443.1	1460.5	100	166.4	1431.2	100
12			20	196.4	1416.5	100	71.9	1374.1	100	261.7	1455.3	100	96.9	1408.4	100
SMT				266.0	1443.0	100				388.0	1467.0	100			
GA				5.0	1218.5	8.0				5.6	1319.9	10.0			

TABLE II. EXPERIMENTAL RESULTS FOR INSTANCES III AND IV.

				INSTANCE III						INSTANCE IV					
				Random			Semi-Random			Random			Semi-Random		
exp	gn	ind	int	t[s]	Q	P	t[s]	Q	P	t[s]	Q	P	t[s]	Q	P
1	8	1	10	13.0	2176.5	6.7	10.5	2077.4	16.7	22.1	2229.5	6.7	19.7	2124.4	33.3
2			20	12.4	2054.3	10.0	10.9	2144.6	16.7	22.0	2193.6	16.7	16.0	2141.2	16.7
3		10	10	121.7	2311.5	46.7	54.8	2217.8	83.3	248.3	2359.1	43.3	101.0	2226.6	70.0
4			20	54.2	2279.4	26.7	**27.6**	**2217.8**	**83.3**	**151.9**	**2353.5**	**43.3**	58.7	2224.4	53.3
5		20	10	324.9	2369.4	76.7	94.3	2284.3	76.7	566.8	2390.7	60.0	195.4	2284.8	90.0
6			20	175.7	2304.2	50.0	58.3	2233.0	70.0	290.8	2334.1	53.3	89.2	2215.2	73.3
7	12	1	10	208.1	2153.4	46.7	55.8	2205.6	90.0	239.7	2216.3	56.7	92.9	2223.2	83.3
8			20	**54.1**	**2274.1**	**36.7**	43.8	2131.7	53.3	64.1	2167.0	26.7	**55.8**	**2226.3**	**60.0**
9		10	10	1727.1	2377.9	100	327.3	2282.2	100	2205.2	2485.3	100	553.7	2328.8	100
10			20	1066.5	2327.7	96.7	213.0	2286.5	100	1325.1	2414.3	96.7	291.6	2246.1	96.7
11		20	10	2814.4	2447.1	100	650.8	2364.7	100	4455.6	2568.2	100	882.4	2408.3	100
12			20	2027.1	2387.3	100	337.8	2317.8	100	2477.0	2469.8	96.7	416.9	2338.2	100
SMT				500.0	2266.0	100				500.0	2409.0	100			
GA				6.0	2085.4	10.0				6.6	2001.9	7.0			

TABLE III. EXPERIMENTAL RESULTS FOR INSTANCES V AND VI

				INSTANCE V						INSTANCE VI					
				Random			Semi-Random			Random			Semi-Random		
exp	gn	ind	int	t[s]	Q	P	t[s]	Q	P	t[s]	Q	P	t[s]	Q	P
1	8	1	10	12.0	560.0	36.7	16.3	546.1	66.7	25.4	584.1	63.3	20.5	572.1	100
2			20	8.8	486.0	16.7	10.8	509.5	66.7	17.4	585.3	33.3	16.4	564.3	90.0
3		10	10	55.0	704.1	93.3	34.1	648.0	96.7	156.1	804.8	100	81.9	699.9	100
4			20	**36.2**	**638.2**	**80.0**	22.7	596.1	93.3	**96.8**	**722.8**	**93.3**	**40.6**	**660.1**	**96.7**
5		20	10	94.9	777.9	96.7	51.8	687.1	96.7	298.7	888.1	100	131.8	783.0	100
6			20	52.0	667.3	83.3	**28.2**	**634.1**	**100**	165.9	808.9	100	81.5	694.4	100
7	12	1	10	69.8	620.2	96.7	31.8	569.6	100	124.4	660.3	100	42.2	546.8	100
8			20	30.4	561.1	93.3	23.0	532.6	96.7	75.1	608.8	90.0	42.5	588.6	96.7
9		10	10	420.0	852.8	100	139.9	731.4	100	1385.4	928.5	100	294.7	769.0	100
10			20	264.1	774.5	100	66.5	620.8	100	619.6	814.4	100	122.3	628.9	100
11		20	10	807.4	935.5	100	275.7	832.7	100	2614.5	993.3	100	643.2	874.4	100
12			20	461.9	852.6	100	100.7	675.7	100	1464.6	927.3	100	260.1	750.9	100
SMT				500.0	781.0	100				500.0	755.0	100			
GA				5.1	436.0	8.0				5.9	537.8	12.0			

III, and V contains 256 offers, which makes the number of the potential solutions equal to $256^{15} = 2^{120}$. In the case of Instance II, IV, and VI, each offer set consists of 512 offers, which results in the search space size as large as $512^{15} = 2^{135}$. The objective functions used for the Instances from I to IV are as follows:

$$Q_{1,2} = \sum_{i=1}^{n} o^i_{j_i,1}, \quad (7)$$

$$Q_{3,4} = \sum_{i=1}^{n} (o^i_{j_i,1} + o^i_{j_i,2}). \quad (8)$$

The set of the constraints of the Instances from I to IV is defined as follows:

$$\mathbb{C}_{1,2,3,4} = \{(o^i_{j_i,2} < o^{i+1}_{j_{i+1},2}) \mid i = 1, \ldots, n-1\}. \quad (9)$$

That is, for the Instances I and II our aim is to maximize a sum of n values, while for the Instances III and IV the sum of $2n$ values is to be maximized. Since the concrete planning is reduced to the constrained optimisation problem and it is clearly separated from the previous planning stages, the concrete planners do not need to "know" what the planning domain is, and what the particular variables mean. This allowed us to generate the instances randomly using our software Offer Generator.

In the case of the Instance V and VI, which are based on Example 1, the objective functions and the constraints are as

follows:

$$Q_{5,6} = \sum_{i=1}^{n} o^i_{j_i,3}, \quad (10)$$

$$\mathbb{C}_{5,6} = \{(o^1_{j_1,1} \leq 100), (o^1_{j_1,2} + o^2_{j_1,2} > 50), (o^i_{j_i,5} = o^{i+1}_{j_{i+1},1}) \mid i = 1, \ldots, n-1\}. \quad (11)$$

This time we can provide a clear interpretation of the objective function and the constraints, as they follow the user query given in the example. The objective functions Q_5 and Q_6 correspond directly to the function Q defined in Example 1. Since the third value of an offer corresponds to the $profit$ attribute, the sum of the profits is to be maximized here. Similarly, the constraint sets also correspond to these defined in Example 1.

Besides the parameters introduced already in Section IV, the standard parameters of GA, used in the hybrid algorithms, have been set to the same values as in the pure GA, that is, the population size – 1000, the number of iterations – $N = 100$, the crossover probability – 95%, and the mutation probability – 0.5%. Moreover, all the experiments with the hybrid algorithms have been performed using $st = 20$, that is, the first *SFII* procedure starts with the 20th iteration. Every instance has been tested 12 times, using a different combination of the remaining parameter values (see Tables from I to III), and every experiment has been repeated 30 times on a standard PC with 2.8GHz CPU and Z3 [25] version 4.3 as the SMT-solving engine.

The results of applying the RH and SRH algorithms to Instances I - VI are presented in Tables I, II, and III, where the columns from left to right display the experiment label, the parameter values, and for each Instance and each hybrid variant the total runtime of the algorithm (**t[s]**), the average quality of the solutions found (**Q**), and the probability of finding a solution (**P**). For reference, we report in the two bottom rows (marked with SMT and GA, respectively) the results of the pure SMT- and GA-based planner. The pure GA-based planner was run with the same parameters values as the hybrid ones. The test has been performed on the same machine.

One can easily see that the quality values obtained in almost every experiment are higher than these returned by GA. However, in several cases either the runtime or the probability is hardly acceptable. On the other hand, for many parameter combinations we obtain significantly better results in terms of the runtime (comparing to the pure SMT) or the probability (in comparison with the pure GA). We marked in bold the results that we find the best for a given instance and a hybrid variant.

Although the results are very promising and encouraging, as one could expect, the hybrid solutions are clearly a trade-off between the three measures: the quality, the probability, and the computation time of the pure algorithms. It is easy to observe that for many parameter valuations the hybrid algorithms outperform each pure planning method provided one or two measures are taken into account only. Moreover, the Semi-Random Hybrid algorithm outranks in almost all cases the Random Hybrid one in terms of the computation time and the probability of finding a solution. On the other hand, since RH runs SMT-solver much more often than SRH, it also finds solutions of better quality than SRH, but at the price of a much longer computation time.

In order to compare the results obtained taking all the three measures into account at the same time, we defined two simple score functions:

$$score_i(P, t, Q) = \frac{P}{t} \cdot (Q - const_i)$$

$$score_{p^2}(P, t, Q) = \frac{P^2 \cdot Q}{t}, \quad (12)$$

where P, t, and Q stand for the probability, the computation time, and the average quality, respectively, and $const_i$ is a parameter, which value is selected for each Instance i from I to VI, to make the scores of the pure GA- and SMT-based algorithm equal. The values of the $const_i$ parameters used for comparing the results for Instances I-VI are as follows: 1150, 1295, 2061, 1906, 386, 514, respectively.

These scores are then selected as the benchmarks for the comparison given in Figure 5. The dark- and light-grey bars correspond to the results obtained with the RH and SRH algorithm, respectively.

The $score_i$ function aims at comparing the results under the assumption that both the pure planning methods are equally effective as far as the three measures are concerned. On the other hand, the $score_{p^2}$ function gives priority to the solutions having a high probability. Obviously, this way, one can define a number of other score functions in order to compare the results according to a personal preference. Notice that another interesting remark can be made about the hybrid parameter values. Namely, the bold values in Tables I, II, and III, as well as the highest chart bars in Figure 5 most often correspond to parameter combinations of the experiment 4, 7, and 8. However, the study of only six instances does not allow us to draw any broad conclusions. Therefore, in our future work we are going to investigate whether these parameter values guarantee to obtain good results in general.

Next, we use the same benchmarks in order to evaluate the efficiency of the IPH algorithm. The results are presented in Table IV. Its first column contains the number of the individuals in the initial population generated by an SMT-solver. While the total population size is equal to 1000, the remaining individuals are created randomly.

The data in the next columns stand for the computation time and the quality of solutions found given for each instance from I to VI.

The most important advantage of the IPH algorithm over the other algorithms is that the probability of finding solutions in all cases is equal to 100%. This is the consequence of generating at least one individual, which is already a solution at the start of GA. Also, the computation time of IPH is much smaller than in the case of the pure SMT as well as the RH and SRH algorithms. On the negative side of the IPH algorithm, the quality function value of the solutions found is lower.

The comparison of the results of all three algorithms based on values of the score functions in shown in Figure 6. At the X axis: $1, 100, 500, 1000$ are the numbers of the individuals generated by the SMT-solver in the initial population; SMT and GA stand for the results obtained using the respective "pure" planning methods while RH and SRH denote the best results obtained with Random and Semi-Random Hybrid algorithms.

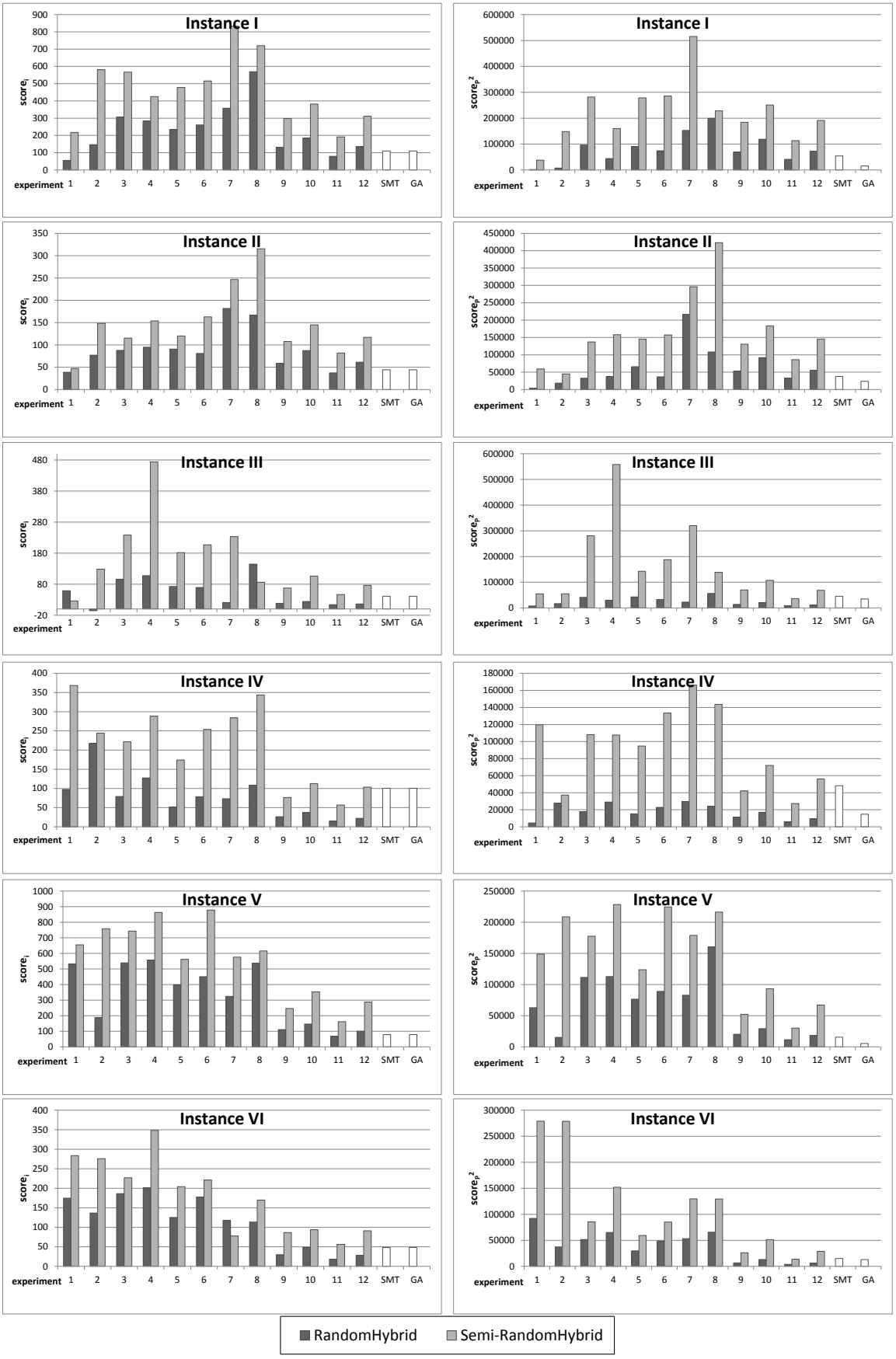


Figure 5. The comparison of the RH and SRH algorithm performance based on two score functions.

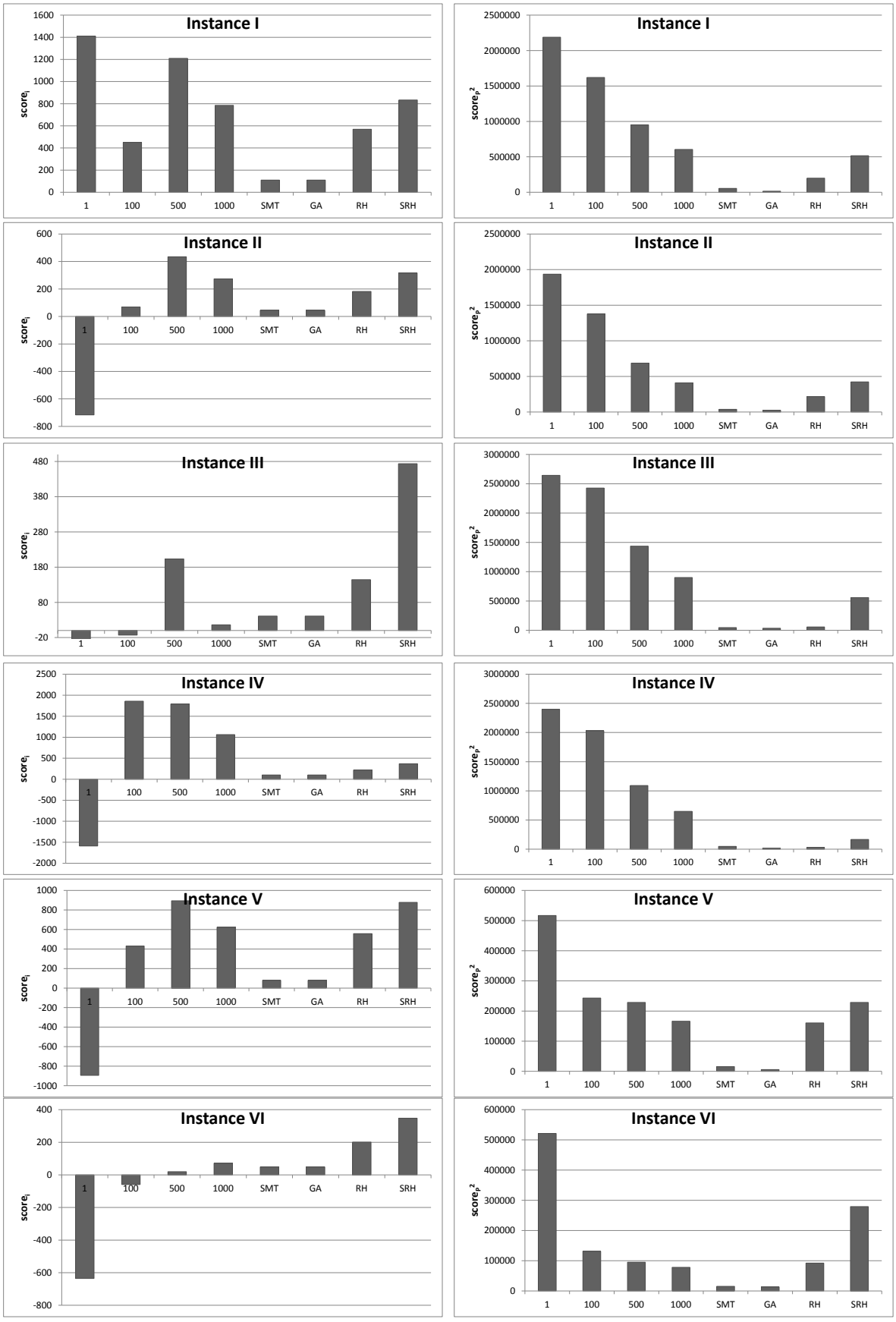


Figure 6. The comparison of the results of the three algorithms based on the values of the score functions.

TABLE IV. EXPERIMENTAL RESULTS OF THE IPH ALGORITHM.

	INSTANCE I		INSTANCE II		INSTANCE III	
inds	t[s]	Q	t[s]	Q	t[s]	Q
1	5,6	1229,5	6,4	1248,8	6,4	1706,8
100	7,3	1183,2	9,5	1301,5	8,4	2059,9
500	13,9	1317,9	20,1	1382,4	14,6	2090,6
1000	21,8	1321,7	33,8	1387,4	23,0	2064,7

	INSTANCE IV		INSTANCE V		INSTANCE VI	
inds	t[s]	Q	t[s]	Q	t[s]	Q
1	7,5	1788,0	6,4	329,1	8,8	458,2
100	10,3	2098,5	19,3	469,2	37,6	491,7
500	20,9	2280,3	27,8	634,5	55,2	524,7
1000	35,2	2280,3	37,2	618,0	72,5	565,6

VI. CONCLUSION AND FUTURE WORK

In this paper three versions of the hybrid concrete planner have been implemented and several experiments have been performed. The experimental results show that even when using a straightforward strategy of combining the SMT- and GA-based approach, one can obtain surprisingly good results. We believe that all of the proposed methods are of high potential. However, the most promising approach seems to be the InitPop Hybrid algorithm, due to its relatively low computation time and the ability to always return a solution, if there exists one. These advantages follow from dividing the tasks of both the combined methods so that each of them is doing its best. While the SMT-solver deals easily even with a large number of constraints, the genetic algorithm copes with the objective functions.

An important task to be addressed in a future work will consist in investigating how to choose the parameter values in order to get a trade-off between the quality, the probability, and the computation time desired by the user. Moreover, using the experience gained from the concrete planning, we intend also to develop a hybrid solution for the abstract planning stage.

ACKNOWLEDGMENT

This work has been supported by the National Science Centre under the grant No. 2011/01/B/ST6/01477.

REFERENCES

[1] A. Niewiadomski, W. Penczek, and J. Skaruz, "Genetic algorithm to the power of SMT: a hybrid approach to web service composition problem," in Service Computation 2014 : The Sixth International Conferences on Advanced Service Computing, 2014, pp. 44–48.

[2] M. Bell, Introduction to Service-Oriented Modeling (SOA): Service Analysis, Design, and Architecture. John Wiley & Sons, 2008.

[3] S. Ambroszkiewicz, "Entish: A language for describing data processing in open distributed systems," Fundam. Inform., vol. 60, no. 1-4, 2003, pp. 41–66.

[4] J. Rao and X. Su, "A survey of automated web service composition methods," in Proc. of SWSWPC'04, ser. LNCS, vol. 3387. Springer, 2005, pp. 43–54.

[5] D. Doliwa et al., "PlanICS - a web service compositon toolset," Fundam. Inform., vol. 112(1), 2011, pp. 47–71.

[6] A. Niewiadomski and W. Penczek, "Towards SMT-based Abstract Planning in PlanICS Ontology," in Proc. of KEOD 2013 International Conference on Knowledge Engineering and Ontology Development, September 2013, pp. 123–131.

[7] A. Niewiadomski, W. Penczek, and J. Skaruz, "SMT vs genetic algorithms: Concrete planning in PlanICS framework," in CS&P, 2013, pp. 309–321.

[8] J. Skaruz, A. Niewiadomski, and W. Penczek, "Automated abstract planning with use of genetic algorithms," in GECCO (Companion), 2013, pp. 129–130.

[9] A. Niewiadomski, W. Penczek, J. Skaruz, M. Szreter, and M. Jarocki, "SMT versus Genetic and OpenOpt Algorithms: Concrete Planning in the PlanICS Framework," (accepted to Fundam. Inform.), 2014.

[10] A. Niewiadomski, W. Penczek, and J. Skaruz, "A hybrid approach to web service composition problem in the PlanICS framework," in Mobile Web Information Systems, ser. Lecture Notes in Computer Science, I. Awan, M. Younas, X. Franch, and C. Quer, Eds. Springer International Publishing, 2014, vol. 8640, pp. 17–28. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10359-4_2

[11] G. Canfora, M. D. Penta, R. Esposito, and M. L. Villani, "An approach for QoS-aware service composition based on genetic algorithms," in Proceedings of the 2005 Conference on Genetic and Evolutionary Computation, 2005, pp. 1069–1075.

[12] Y. Wu and X. Wang, "Applying multi-objective genetic algorithms to QoS-aware web service global selection," Advances in Information Sciences and Service Sciences, vol. 3(11), 2011, pp. 134–144.

[13] M. AllamehAmiri and V. D. dn M. Ghasemzadeh, "Qos -based web service composition based on genetic algotihm," Journal of AI and Data Mining, vol. 1, 2013, pp. 63–73.

[14] Y. Y. Zhang, H. L. Xiong, and Y. C. Zhang, "An improved genetic algorithm of web services composition with qos," Advanced Materials Research, vol. 532, 2012, pp. 1836–1840.

[15] T. Weise, S. Bleul, and K. Geihs, "Web service composition systems for the web service challenge – a detailed review," 2007.

[16] H. Jiang, X. Yang, K. Yin, S. Zhang, and J. A. Cristoforo, "Multi-path QoS-aware web service composition using variable length chromosome genetic algorithm," Information Technology Journal, vol. 10, 2011, pp. 113–119.

[17] L. Zhang, B. Li, T. Chao, and H. Chang, "On demand web services-based business process composition," in Proc. of the IEEE Int. Conf. on Systems, Man and Cybernetics. IEEE Computer Society, 2003, pp. 4057–4064.

[18] D. B. Claro, P. Albers, and J. kao Hao, "Selecting web services for optimal composition," in Proc. of the 2nd Int. Workshop On Semantic And Dynamic Web Processes, 2005, pp. 32–45.

[19] I. Salomie, M. Vlad, V. Chifu, and C. Pop, "Hybrid immune-inspired method for selecting the optimal or a near-optimal service composition," in Proc. of the Computer Science and Information Systems. IEEE Computer Society, 2011, pp. 997–1003.

[20] J. Xu and S. Reiff-Marganiec, "Towards heuristic web services composition using immune algorithm," in Proc. of the IEEE Conf. on Web Services. IEEE Computer Society, 2008, pp. 238–245.

[21] C. Hu, X. Chen, and X. Liang, "Dynamic services selection algorithm in web services composition supporting cross-enterprises collaboration," Journal of Central South University of Technology, vol. 16, 2009, pp. 269–274.

[22] J. A. Parejo, P. Fernandez, and A. R. Cortes, "Qos-aware services composition using tabu search and hybrid genetic algorithms." Actas de los Talleres de las Jornadas de Ingenieria del Software y Bases de Datos, vol. 2(1), 2008, pp. 55–66.

[23] Z. Jang, C. Shang, Q. Liu, and C. Zhao, "A dynamic web services composition algorithm based on the combination of ant colony algorithm and genetic algorithm," Journal of Computational Information Systems, vol. 6(8), 2010, pp. 2617–2622.

[24] J. Skaruz, A. Niewiadomski, and W. Penczek, "Evolutionary algorithms for abstract planning," in PPAM (1), ser. Lecture Notes in Computer Science, R. Wyrzykowski, J. Dongarra, K. Karczewski, and J. Wasniewski, Eds., vol. 8384. Springer, 2013, pp. 392–401.

[25] L. M. de Moura and N. Bjørner, "Z3: An efficient SMT solver," in Proc. of TACAS'08, ser. LNCS, vol. 4963. Springer-Verlag, 2008, pp. 337–340.

A Combined Simulation and Test Case Generation Strategy for Self-Adaptive Systems

Georg Püschel, Christian Piechnick, Sebastian Götz,
Christoph Seidl, Sebastian Richly, Thomas Schlegel, and Uwe Aßmann
Software Technology Group, Technische Universität Dresden
Email: {georg.pueschel, christian.piechnick, sebastian.goetz1,
christoph.seidl, sebastian.richly, thomas.schlegel, uwe.assmann}@tu-dresden.de

***Abstract*—With the introduction of self-adaptivity in software architecture, it becomes feasible to automate tasks that are performed under changing conditions. In order to validate systems with such capabilities, the conditions have to be enforced and reactions verified. An adequate set of scenarios must be performed to assure the required quality level. In our previous work, we investigated a set of requirements for a self-adaptive system validation strategy as well as a high-level solution scheme. In this paper, we instantiate this scheme and propose a set of timed models that work together as black box test model for our example SAS HomeTurtle. The model can be either used for simulation or test case generation; for both approaches, a unifying infrastructure is described. We further show an example simulation run and present our implementation—the Model-driven Adaptivity Test Environment. The proposed methodology enables test experts to maintain the complex behavior of SAS and cover an adequate part of it in testing.**

Keywords*—*Self-Adaptive Systems; Service Robots; Model-Based Testing; Simulation; Feedback Loops

I. INTRODUCTION

In our original work [1], we introduced a strategy for creation and execution of timed simulation models for Self-adaptive Systems (SAS). This kind of system adapts itself according to changes in its environment [2][3]. The continuous execution of sensor monitoring, analysis, planning, and adaptation execution is organized in feedback loops [4]. Due to the use of intelligent reasoning strategies, an SAS is capable of fulfilling its tasks more efficiently or it even may find solutions to tasks that were not explicitly defined at design time. Potential adaptations encompass simple changes of certain control variables, structural re-organization of components and the exchange of behavioral strategies that might better fit for the found environment situations.

In our work, we aim to provide solid SAS development methods and, thus, we also require a validation approach that is able to deal with the complexity of such self-adaptive behavior. The mechanisms that decide autonomously have to be validated extensively before deploying the system in a productive environment. A limitation constitutes from the fact that an SAS can be adapted externally or reason about unanticipated events. These aspects can never be tested comprehensively before delivery and, thus, are excluded from the scope of our proposed solution.

However, even for these systems, the user's trust has to be gained by examining the system in an appropriate variety of scenarios. Hence, validation methods can be performed on different abstraction layers as, for instance, the German V-Modell [5] proposes. On the lowest abstraction layer of modules, knowledge of code and design models can be utilized. However, due to the complexity and large variety of possible situations, performing a comprehensive validation (e.g., by deriving and executing test cases) on these levels is expensive.

In contrast, validating SAS applications on acceptance level, based on requirements of a more abstract specification, is more promising. For this purpose, the engineer no longer relies on detailed knowledge of the system interior but on a black box interface that is used to enforce situations and verify the outcome. In contrast to white box testing, black box approaches cannot locate faults. Thus, each found failure has to be analyzed by additional means in a subsequent step.

Setting up a black box interface that provides all necessary operations to interact with the system and to query information that has to be examined is the first crucial task during the validation phase. The expected behavior of the SUT can be specified based on this interface. An appropriate method for such specification is model-based testing (MBT, [6]). In this approach, a test model is specified and test cases are generated from it. In the most comprehensive variant of MBT, the model captures all information about which test data is sent to the tested system and which reactions are expected. In this way, the test model serves as a *test oracle*, which determines the correctness of observed reactions or predicts these reactions.

A further problem can arise when the SAS is deployed in complex environments where not every property of a situation can be enforced. For instance, the interaction with certain entities (e.g., hardware controllers or physical objects) is difficult to formalize. Instead, the test model designer may specify some future decisions depending on the state that is observed from these entities at test execution time (i.e., a run-time-dynamic oracle). Test cases do not support decisions on run-time information, as the generated actions cannot be exchanged or reordered in case of adaptations depending on such properties. Instead, the test model has to be executed directly (i.e., without test case generation). Therefore, we propose to perform simulation and capture the discussed non-specifiable parts of the system or test environment *"in-the-loop"*. In our concept simulation means to produce inputs that are given to the real SUT and the test model and compare the results of both. A drawback of simulation in comparison to the test case generation is that there is no fixed set of test cases to be replayed for regression. In consequence, both test case generation or simulation may be employed depending on the quality requirements and relevant context of a set of system parts under test.

However, both methods rely on a common artifact—a test or simulation model. A generic SAS testing framework has to provide a respective metamodel that is expressive enough for compact specification of all behavioral and adaptation-related

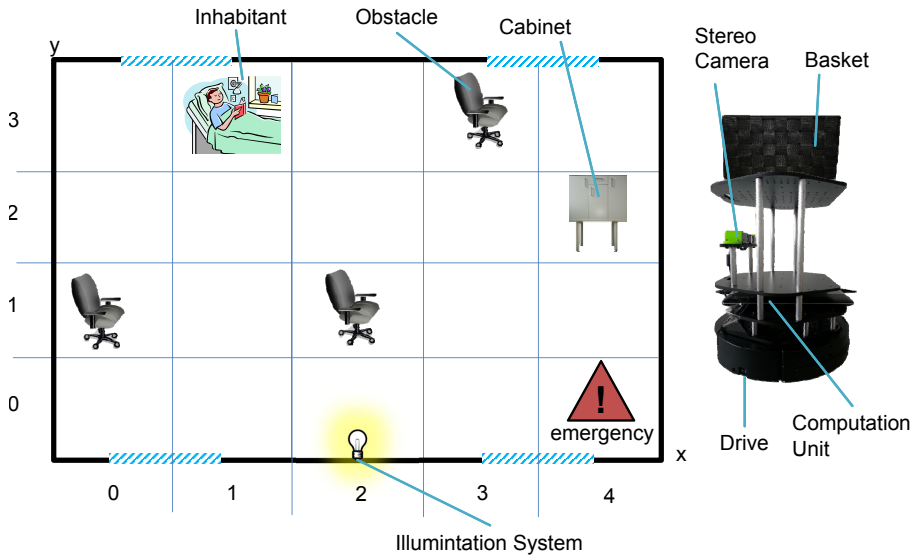


Fig. 1. Scenario: HomeTurtle operating in a flat.

aspects. These aspects are given by several requirements that we derived in our previous work [7]. The following requirements were formulated as goals:

(1) Correct sensor interpretation
(2) Correct adaptation initiation
(3) Correct adaptation planning
(4) Consistent adaptation/system interaction
(5) Consistent adaptation execution
(6) Correct system behavior (especially actuator actions)

Goals (1) and (6) include the validation of the correctness in sensor perception and actuator control. Both properties can be checked in isolation by instrumenting the respective drivers. In this paper, we focus on the goals (2)-(5), which directly deal with the SAS feedback loop (sometimes referred to as MAPE loop: monitor, analyze, plan, execute [4]). In order to match the requirements, the test metamodel has to provide means for defining in which situations an adaptation has to be initiated (goal 2), how the system is expected to adapt (goal 3), how the adaptation is expected to be scheduled with non-adaptation-related behavior (goal 4), and how the result of the adaptation must look like (goal 5).

In [1], we proposed a methodology to separate all these aspects in components of a composite simulation model. Parts of our model are enriched with *assertions* on the System Under Test's (SUT) interface in order to define how a simulation state has to be verified. The complete modeling methodology is illustrated using our *HomeTurtle* domestic robot application. Throughout the paper, the Unified Modeling Language (UML) and Object Constraint Language (OCL) are used for representing almost all details of the model by a widely-understood standard syntax and semantics.

In the HomeTurtle scenario, a robot is deployed in a flat of a handicapped person and is capable of delivering various items, which are stored in a software-controlled cabinet. Besides reciting this illustrative example as well as the introduced test methodology, we contribute the following aspects in this paper:

1) **Simulation- vs. generation-based validation:** We describe how the proposed modeling concepts are used for simulation alternatively to test case generation. For this purpose, an infrastructure is proposed that unifies both approaches.

2) **Details on implementation:** All concepts have been implemented in our integrated test environment MATE. The components of this tool are presented.

3) **Extended related work:** We extend the discussion on the body of knowledge in SAS testing.

The remainder of this paper is structured as follows: In Section II, we introduce our example adaptive system. In Section III, we present our approach based on this example. In Section V, we illustrate an example simulation run. In Section IV, we describe how the necessary infrastructure for simulation and generation can be unified. In Section VI, we present our implementation and experimental environment. In Section VII, we discuss related work. In Section VIII, we discuss conclusion and future work.

II. Example Application: Home Turtle

In this section, we present an illustrative example of an SAS controlling a robot that is instructed to support a handicapped person at home. The scenario is depicted in Figure 1. A service robot "HomeTurtle" (an extended version of the TurtleBot platform [8]) is initially deployed in the flat. The task of the robot is to locate and deliver a desired item to the user (i.e., the inhabitant). Respective items can be dropped from a *cabinet* into a basket mounted on top of the robot. For this purpose, the cabinet contains several boxes with magnetically clamped flaps. The magnets are triggered from a WiFi-enabled embedded device.

Initially, an user instructs the robot by entering the desired item (e.g., "towel") using a Tablet PC that is accessible nearby. Using a wireless network, the robot can query the flat's map,

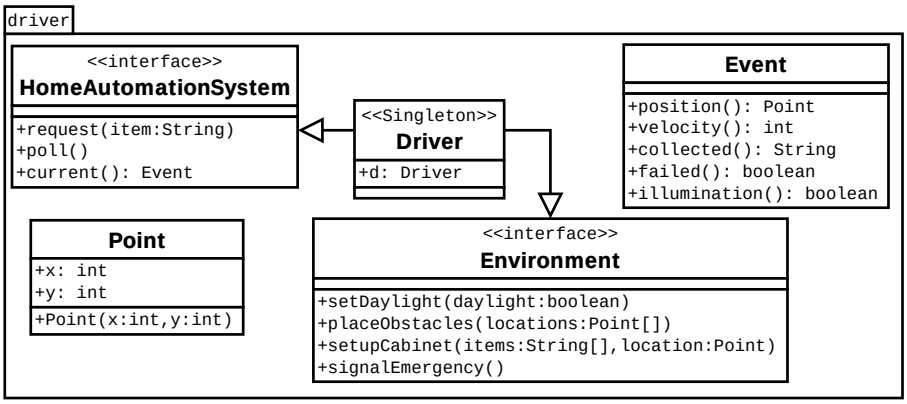


Fig. 2. Test driver interface.

available cabinets including their positions and contents. After this information was gathered, the robot is able to inform the user whether the desired item is available. Once the item was located, a route is planned and the robot starts driving. In this process, the robot has to avoid collisions with walls and other obstacles (symbolized by office chairs). After approaching a cabinet and parking in a predefined position underneath it, the robot signals the cabinet to drop the requested item. Afterwards, it drives back to the user. During the complete process, the environment may signal an emergency (e.g., a fire or medical emergency). In this situation, the robot is expected to drive to its emergency position as labeled in our illustration. Thus, it avoids obstructing access of human helpers to the inhabitant.

The following sensors and actuators are used to accomplish the robot's task:

- **Robot drive:** The robot drive has three modes for stopping (0=stop) and driving in arbitrary directions with two different velocities (1=slow, 2=fast).
- **Stereo camera:** Can be used to recognize walls and obstacles.
- **On-board computation unit:** The robot runs its operations on-board using a fix-installed netbook that connects to all the hardware on the robot.
- **Smart illumination system:** The flat is equipped with room lights that can be operated by the software system to improve the flat's illumination on demand. In this way, the object recognition performed using the stereo camera is supported.
- **Local WiFi:** The robot as well as the cabinet are connected to a wireless network. Thus, the flat's map and information about the cabinet's position and contents can be shared.

Furthermore, to improve its behavior, assure safety and minimize operation time, the following *adaptations* are possible:

- **Improve illumination:** If the robot enters a room and daylight from the windows is not sufficient for object recognition, the robot connects to the illumination system and activates it. After delivery, the supporting illumination is switched off again.
- **Location-dependent velocity:** While driving at fast mode velocity, the robot is not able to stop in time if an obstacle is detected. As the obstacles' positions may change, the robot is expected to run in slow mode during the current request as long as the current position was not explored during this request.

In order to send input data to the real system and to verify its output during simulation or test case execution, a *test driver* is required. For our example, we implemented such a driver whose interface is depicted as UML Class Diagram in Figure 2. The class `Driver` holds a static instance `Driver.d` and implements two interfaces: Firstly, `Environment` provides methods to enforce an emergency signal, mock a light state, and setup obstacles and a cabinet. In order to reduce the scenario's complexity, we assume that the positions of the inhabitant and emergency locations as well as the room's layout are static.

Secondly, the interface `HomeAutomationSystem` can be used to request a new item for delivery or to retrieve events that can be verified during simulation. The driver's event-based architecture allows verifying changes of the system without surveying it actively during the whole test execution. Changes in the environment can be tracked by investigating multiple events within one verification action. Therefore, events only have to be produced when the environment changes. Each instance of class `Event` captures information about the current position, velocity, and illumination. It also informs whether an item was collected or the search has failed.

Only a subset of the driver's functionality can be automated. Especially, for obstacle placement and cabinet setup, a dialog is shown to the test engineer, which lists instructions on necessary manual manipulations. All other functions are implemented using the system's sensors (brightness, velocity, etc.) and a wireless-switchable light bulb for change of illumination.

III. Validating SAS by Using an Adaptive Simulation Model

In this section, we present our methodology. The briefly discussed challenges are tackled in different components of a black box simulation model. These components, as well as their dependencies, are depicted in Figure 3. Each component matches a set of specific concerns that were separated in order to decouple the responsibilities during the design process. The

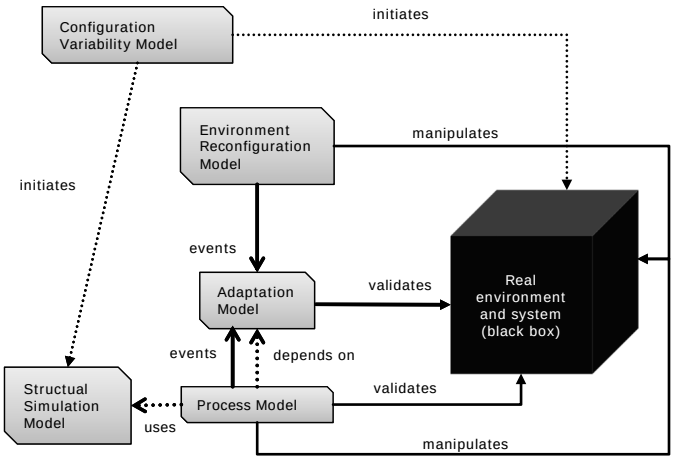


Fig. 3. Concern-separated components of the simulation model.

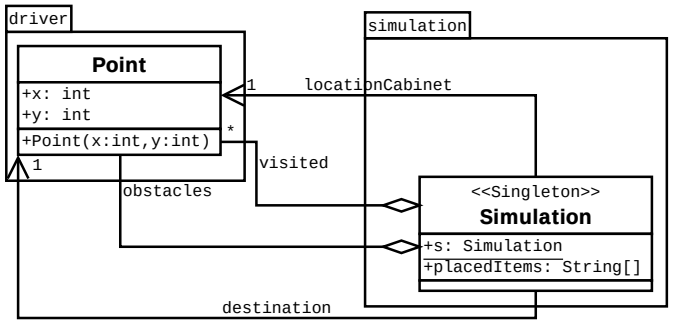


Fig. 4. Structural simulation model.

model is as much as possible based on Unified Modeling Language (UML) 2 [9], Object Constraint Language (OCL) [10] and a special version of equivalence class trees [11]. Actions are formulated as Java method calls.

The current state of the test execution is represented by the *Structural Simulation Model* (i.e., a UML class model). Based this state, the operational model of the running test scenario is given by the *Process Model* that is represented as state chart. The actions performed during execution are, firstly, the requests that are sent to the test driver and, secondly, *assertions* that determine whether the received events are correct in the current state. Thus, the state of the simulation model represents assumptions on the state of the real system. During the initiation of the system, the environment is set up and, synchronously, the Structural Simulation Model is configured with information that reflects this initial environment setting. As there may be different variants of initial configurations, the *Environment Variability Models* defines an equivalence class tree that allows to derive such configurations. The *Environment Reconfiguration Model* contains state charts with actions that define environment manipulations in order to trigger adaptation in the real system. As it defines an operational order of manipulations, requirement (3)–*correct adaptation planning*— can be dealt with. Regarding the requirement (2) (cf. Section I), it has to be validated whether system correctly adapts to these changes. Therefore, the Environment Reconfiguration Model produces events that are consumed by an *Adaptation Model* that reflects adaptation modes and validates them using assertions (requirement (5)–*consistent adaptation execution*). This Adaptation Model is a state charts as well. Events can also be produced by the Process Model and its behavior can be tailored to the Adaptation Model's state. Thus, requirement goal (4)–*consistent adaptation/system interaction*–is matched.

Basically, data flow between all model components follows the Counter Feedback Loop (CFL) that we claimed to be a central requirement to SAS test approaches in [12]. CFL proposes that a test system has to work vice versa to SAS feedback loops: Instead of monitoring the environment and deducing adaption decisions, an CFL-based based test workflow triggers actively manipulations on environment properties and monitors the SUT's reaction. CFL separates the task of a test system into four periodically-executed steps that are all matched within the proposed components:

1) **Change:** Environment configuration variability model and environment reconfiguration model explicitly define how environment properties can be changed in order to stress to system.

2) **Causal connection:** By exchanging symptom events between environment reconfiguration model and adaption models, the causal connection between change and self-adaption is modeled such that state-dependent adaptations can be verified.

3) **Adaptation plans:** Within the adaption models, accepted events can trigger multiple actions that can be used to describe what parts of the system are expected to be adapted and how. The adaptation outcome must be able to be monitored using the test interface.

4) **Service specification:** In the process model it has to be specified how the performed services of the SUT behaves from a black box perspective according to the reached adaption mode.

In summary, all of our components are designed along the CFL. The details of the individual model components are explained in the following.

A. Structural Simulation Model

During the simulation, several assumptions on the real system have to be managed that are represented by a simulation state. For our example application, the locations of obstacles and the cabinet has be remembered as well as the locations that were already visited. This state is captured by a structural model as depicted in Figure 4. The singleton object `SimulationState.s` holds attributes and aggregates objects that can be manipulated or evaluated by the central Process Model. All (two-dimensional) positions are stored in form of instances of class `Point`.

B. Process Model

The *Process Model* defines the task-specific behavior of the system and how it interacts with its adaptation feedback loops. For our example, we defined these aspects in an

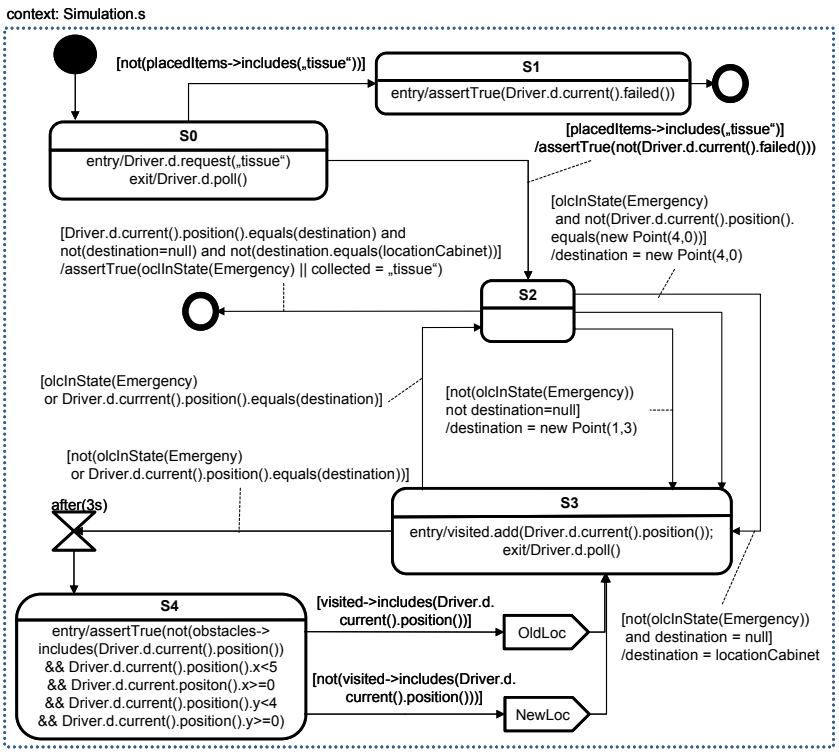


Fig. 5. Process model.

UML State Chart as depicted in Figure 5. The representation uses OCL constraints whose context is the static instance `Simulation.s`. In state `S0`, a request for a towel is initiated and the first event is polled. If the initial configuration set up the cabinet with the desired item, `S1` is reached, otherwise `S2`. The action of the latter transition (i.e., the entry action of `S1`) performs an assertion on whether the real system has either failed or not. If any assertion in the models fails, the simulation is cancelled and an error is signaled. Starting from state `S2`, the robot's destination is determined by evaluating the previous destination value (either null, the start place, the cabinet's place or the emergency position).

States `S3` and `S4` form a feedback loop. When entering `S3`, the current position is appended to the list of visited locations and the next event is polled. In the next step, the loop sleeps three seconds (indicated by the `AcceptTimeAction`, cf. UML spec. [9]). Thus, the Adaptation Models are expected to enforce changes to the environment that are interleaved with the process. Subsequently, in `S4` an assertion is performed in order to ensure no obstacle has been hit and the robot did not leave the boundaries of the scenario. Depending on whether the current position is contained in the `visited` collection, a signal `OldLoc` or `NewLoc` is produced. Therefore, we use the `SendSignalAction` UML element. These signal events are later used to synchronize with the adaptation models. At this point, the feedback loop is restarted. As soon as the destination is reached, the transition to state `S2` is triggered. Another exit possibility from the loop is triggered when the `Emergency` adaptation mode is active. This information can be queried by the `oclInState(...)` function, which is applied to the Adaptation Models. In this way, an interaction between the task-related process and the adaptation mode of the SAS can be modeled. The final state is enabled if the robot reaches a destination that is not the location of the cabinet. The respective transition checks an assertion whether either an emergency was signaled or the correct item was collected.

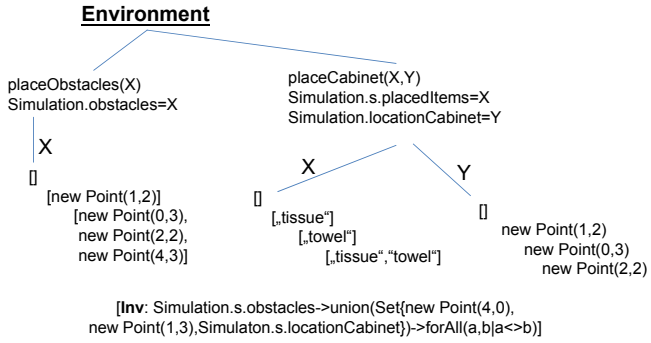


Fig. 6. Environment configuration variability model.

C. Environment Configuration Variability Model

The state space of an environment situation can be enormously large. In testing, this problem is usually dealt by using classification. For instance, data ranges of the system's input parameters are split into equivalence classes and only representatives are tested. All representatives of an equivalence class are assumed to produce the same output. For our example, we designed a dedicated model as depicted in Figure 6. The hierarchical structure serves as a decision tree for determining under which initial conditions a simulation can be started.

Each one of the `Environment` child nodes performs multiple operations: Firstly, the real system is initiated (e.g., the robot is set up in its initial location) and secondly, the simulation state is manipulated such that it reflects this initial configuration. The operations are parameterized with one or two substitution variables. Each variable can be replaced by one of the concrete values in its leaf nodes. The latter ones are the equivalence class representatives. Furthermore, the model contains an invariant to prohibit configurations where the robot's start position, obstacles, or the cabinet are put in the same location.

Basically, this model represents the variability of possible environment settings. Thus, more sophisticated models of variability (e.g., attributed feature models [13]) can also be used for the same purpose. Inherent invariants of such models can restrict the configuration variability space to a manageable size. However, a specific challenge of SAS is to validate whether the system adapts correctly to changes of this configuration. Therefore, in the next section, the configurations dynamics are defined.

D. Environment Reconfiguration Models

The left part of Figure 7 depicts a model of environment reconfiguration. In the upper chart, the entry point of the first state sets the environment daylight to `true`. The driver is now in charge of mocking the brightness sensor's input data and, thus ,enforces the system to adapt. In order to reflect the expected adaptation in the simulation model, a signal `Day` is produced that later will be received by the Adaptation Model. After five seconds, the daylight setting is inverted and the `Night` signal is sent. After additional two seconds, the reconfiguration loop restarts. The lower chart performs a loop that every three seconds demands the simulation to decide of an emergency is signaled or not. This decision can, for instance, be determined randomly or by the user.

Using such environment reconfiguration models, scenarios with different operational orders can be generated. Based on these scenarios, the SUT is stressed and its reactions are exhaustively validated. Using timing, the variety of interleaving possibilities with actions from the Process Model can be reduced.

E. Adaptation Model

Adaptation models define how a configuration has to be altered in response to a received signal. Signals are produced by either the Environment Reconfiguration Models or by the Process Model in order to notify about a condition that may cause an adaptation. The left part of Figure 7 depicts three state charts for the velocity, illumination, and emergency adaptations.

States of an adaptation state chart may contain an entry operation, which performs a validation on the system's adaptation mode. Using UML `AcceptEventActions`, the automaton is designed to wait for the signals. After a signal was received, a new system event is retrieved (`poll()`) such that the assertion is performed on a fresh information basis. Each Adaptation Model stores a specific aspect of the SUT's adaptation mode. Behavioral adaptations are defined using constraints on the Adaptation Models' states.

IV. A Common Infrastructure for Simulation and Test Case Generation

The constructed model can be used for both test case generation and simulation. For test case generation, generators perform a *reachability analysis*, which produces a reachability tree. The tree's root node represents the initial system's state; child states can be discovered by state-changing actions. Each path from the tree's root to a terminal leaf (i.e., where no new actions can be performed) forms a unique test case. The depth of this tree can not only be enormously large but also potentially infinite due to loops and actions without conditions within the modeled control flow. Furthermore, the tree's breadth grows with the degree of indeterminism in each state (i.e., the number of child states respectively applicable actions). Thus, an adequacy criterion is applied that restricts the number of deducible test cases by certain kinds of *adequacy criteria* in form of quantitative measures on the model's elements (e.g., number of states or transitions) or the resulting test cases (e.g., number or length). The benefit of the generation-based approach is that the generated test suite can be re-run for new versions of the examined SAS in the sense of regression testing. As a result, test coverage and test results can be compared quantitatively.

Alternatively, in simulation, no fixed test suite is maintained. In comparison to the generation approach, only one path through the reachability tree is traversed. Therefore, the model is directly executed by an interpreter. If multiple actions can be performed in a single state (a path branches), this indeterminism can be resolved by a human tester or an automatic mechanism (e.g., a heuristic based on a coverage criterion). The major advantage of simulation over generation is the ability the react to change of sensed environment properties. This capability can be used for elements, whose behavior cannot be modeled as precisely as necessary for predicting output. For instance, the movement of physical objects through space is such dynamic that steering it according to generated test data is too expensive or even not possible at all. If the tested system monitors the object's location and the expected reaction to this property has to be tested, it is more effective to deploy the real object, monitor its location and use this data as input for the prediction of expected reaction. In this manner, the real object is taken "in the loop". In generated test cases, a reaction to such properties is not possible as assertions cannot be defined depending on past observations.

For simulation, the metamodel only requires a small set of additional concepts. The model must allow computing test input from monitored data of the in-the-loop objects and for verifying test output based on this run-time information. Therefore, access to query operations of the test interface has to be provided. The rest of the model's capabilities is equivalent for simulation.

As presented, both approaches have their pros and cons. Hence, we constructed an infrastructure that enables test engineers to make use of both methods. Figure 8 depicts involved artifacts, data flow and processing actions in this infrastructure. SUT and environment are accessible via a test interface as presented in Section II. The SUT was built from a design model, which was constructed according to a set of requirements. The requirements document is also the foundation for the validation model (i.e., the test or simulation model). If certain environment objects have to be tested in the loop, query calls to the interface have to be embedded to the validation

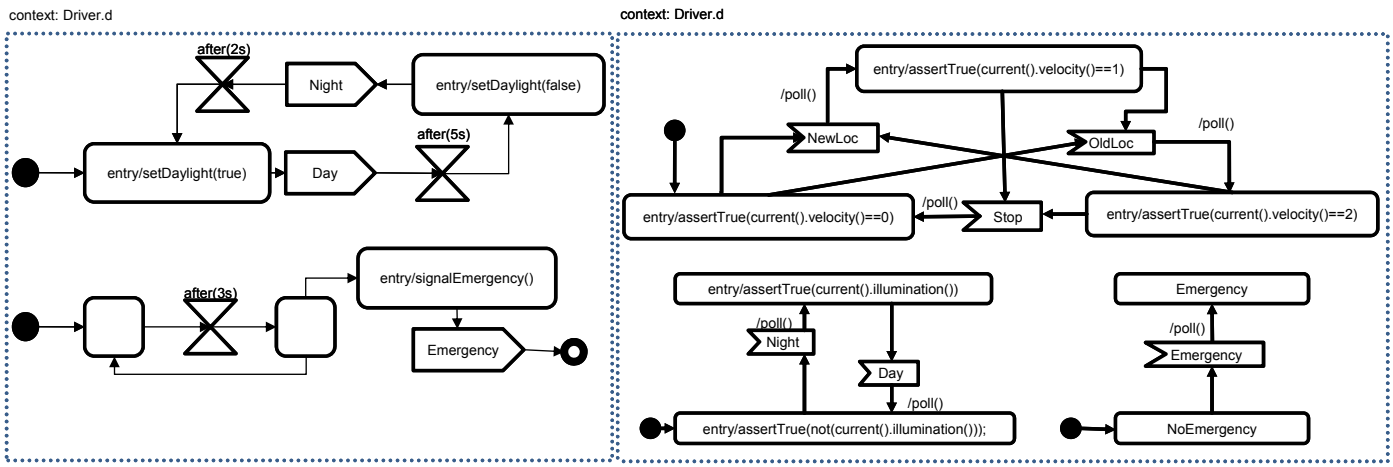


Fig. 7. Left: Environment reconfiguration model. Right: Adaptation models.

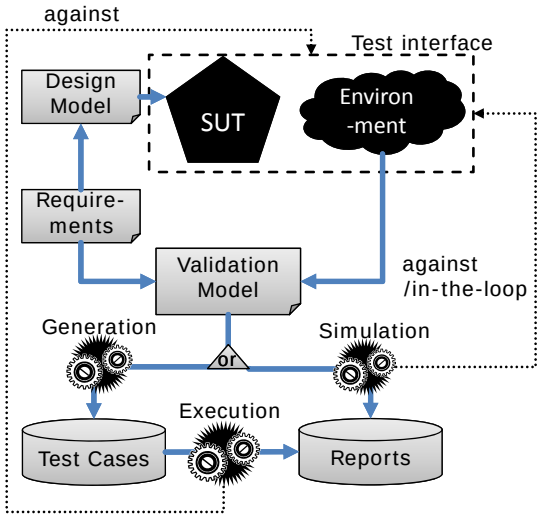


Fig. 8. Simulation and generation infrastructure.

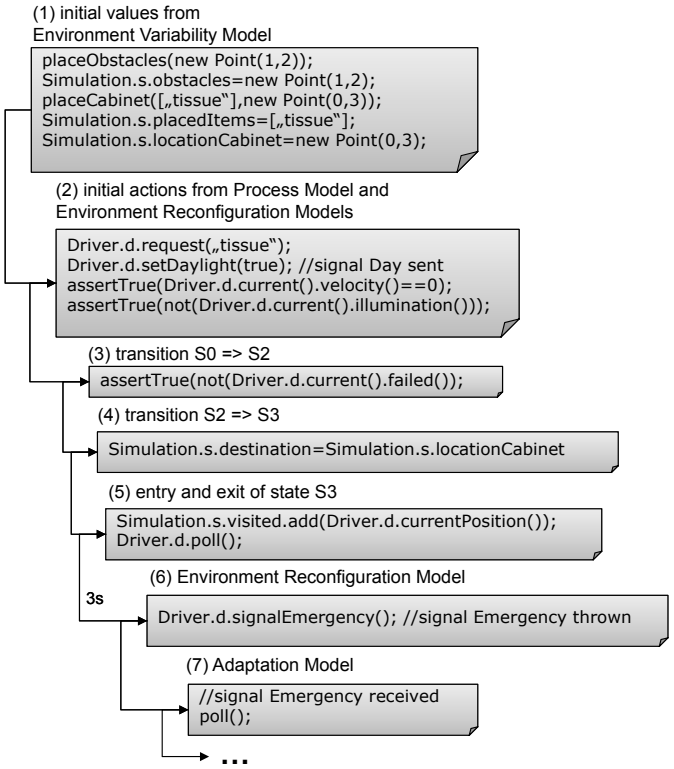


Fig. 9. Excerpt of an example simulation run.

model as well. After specifying the model, it is given as input to either the generator or the simulation engine. The generator produces a set of test cases, which can be run manually or automatically against the exposed test interface. The simulator instead only traverses one trajectory through the state space. Both types of execution results contain the list of performed steps as well as verdicts (e.g., PASS, ERROR, INCONCLUSIVE). In this way, the output of both approaches can be utilized in the following quality improvement.

V. Simulation Example Run

To clarify the models' interactions, we illustrate an excerpt of an example simulation run in Figure 9. The simulation is indeterministic as there can be several execution paths. Sequence (1) of operations is generated by the Environment Variability Model. The simulator automatically selects a solution of the model's invariant such that no obstacle position equals the positions of the inhabitant, cabinet, or emergency stop. When the different state charts are initiated, operations sequence (2) is performed as defined in the initial states. When the Environment Reconfiguration Model sets the daylight property, a signal `Day` is produced. However, as the respective Adaptation Model has no matching outgoing transitions in its initial state, this signal is ignored in this specific state. Sequences (3) and (4) are generated when the transitions `S0->S2` and `S2->S3` are triggered. `S0->S1` cannot be executed as the tissue item was placed in the cabinet during operation of sequence (1). Subsequently, in sequence (5) the entry and exit action of `S3` are executed. After this point, the Process Model waits for three seconds as defined and, consequently, there is an indeterministic decision point in the Environment Reconfiguration Model where either an emergency is signaled or not. We assume that the

simulation determines to generate the emergency such that in sequence (6), the driver is called and the respective signal is produced. In sequence (7), the Adaptation Model receives this signal and switches to the emergency mode after polling a new event. Afterwards, the simulation starts validating whether the robot correctly drives to the emergency stop.

For test generation, all possible trajectories through the state space would be searched and saved as test cases. The number of these test cases is then restricted by a test adequacy criterion such as state or transition coverage. The model of our example is not appropriate for test case generation as it introduces decisions based on runtime information. In particular, the position of the robot is not predicted but constantly queried as it cannot be modeled with an adequate precision. Therefore, the location property has been taken ”in-the-loop” here.

VI. Implementation and Experimental Environment

Syntax and semantics of all used models were implemented in our *Model-driven Adaptivity Test Environment* (MATE). Figure 10 shows a screenshot of its graphical user interface. MATE is an integrated test environment including the following components:

Metamodel implementation: All models proposed here require a metamodel containing concepts to be instantiated. For this purpose, the Eclipse Modeling Framework (EMF, [14]) was used. Besides UML and OCL, a metamodel for creating instances of environment variability models was required.

Model editors: Model construction is enabled by a set of graphical editors. These editors not only support drawing UML and the variability model but also include parsing of the textual elements into their abstract syntax (cf. Figure 10, marking ①).

Test case generator: Using the created models, test cases can be generated. Therefore, an interpreter implements the metamodel's semantics, traverses through the state space and produces one test case for each termination reached. In order to restrict the generation time, the maximal number of test cases can be specified as well as different adequacy criteria.

Interactive simulator: In order to run simulations, the interpreter can be run synchronously with the system instead of generation. Decisions can be either delegated to a heuristic algorithm or performed manually. Heuristics can be implemented project-specific using an appropriate interface. During simulation, the user can visualize the current execution state within the model editors and inspect the state's constituting variable assignments. The simulator's user interface shows a reachability tree, which can be inspected as well (cf. Figure 10, marking ②). Thus, the tester can interact with the interpreter and find situations where variability multiplies the number of sub-paths.

Automation framework: As our approach and tooling is designed to be generic for SAS, it must be reusable. Due to this reason, a framework is provided that allows mapping of actions and queries to a concrete SAS platform. This framework can be used for both executing generated test cases and simulation.

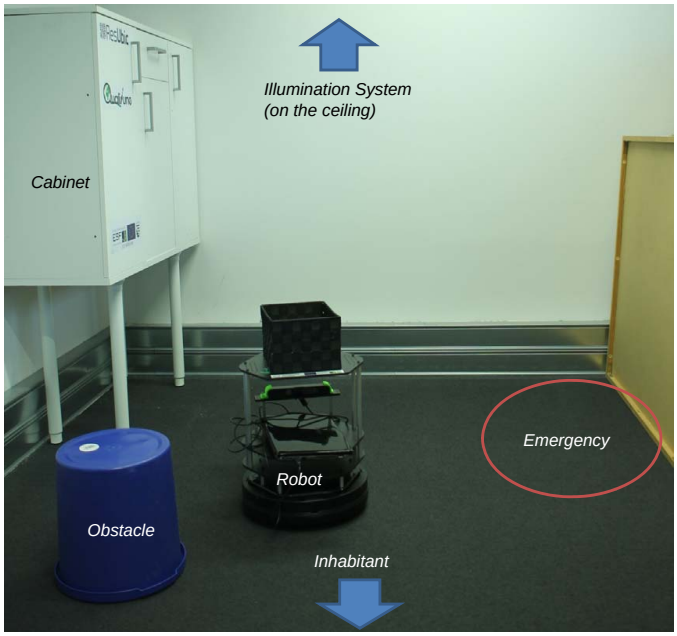


Fig. 11. The HomeTurtle lab.

Reporting tools: The system's quality can be evaluated statistically by reports exported from either executed test cases or simulation runs. Reports themselves are model-based and include verdicts as PASS, FAIL, ERROR, or INCONCLUSIVE. The relation of these verdicts among a set of test case reports can be visualized in appropriate bar diagrams (cf. Figure 10, marking ③).

All these tooling components allow engineers to perform the complete test modeling, execution, and reporting process within a single integrated test environment.

In our previous work, we developed the Smart Application Grid (SMAG) framework that can be used for architectural run-time adaptation [15]. Based on SMAG, we created the self-adaptive HomeTurtle software. An impression of the physical experimental environment is given in Figure 11. In order to show the feasibility of our validation approach, a platform-specific HomeTurtle test driver was developed as well. It directs the operation calls produced by the model to the real system and, vice versa, generates events from the system's observed behavior. However, not every modeled operation can be performed automatically. The initial configuration of the environment (setting up the cabinet's content, placing obstacles, etc.) and the validation whether the correct item was collected are performed manually by the test engineer. During phases, in which test execution had to be automated, the validation directly benefits from the model-driven nature of our approach. Its advantage in manually performed action is given by the reproducibility of simulation paths. If any path fails during a test, it can be recorded, analyzed and even be re-executed later on.

VII. Related Work

Comprehensive validation approaches for self-adaptive systems are still rare in literature. According to Salehie et

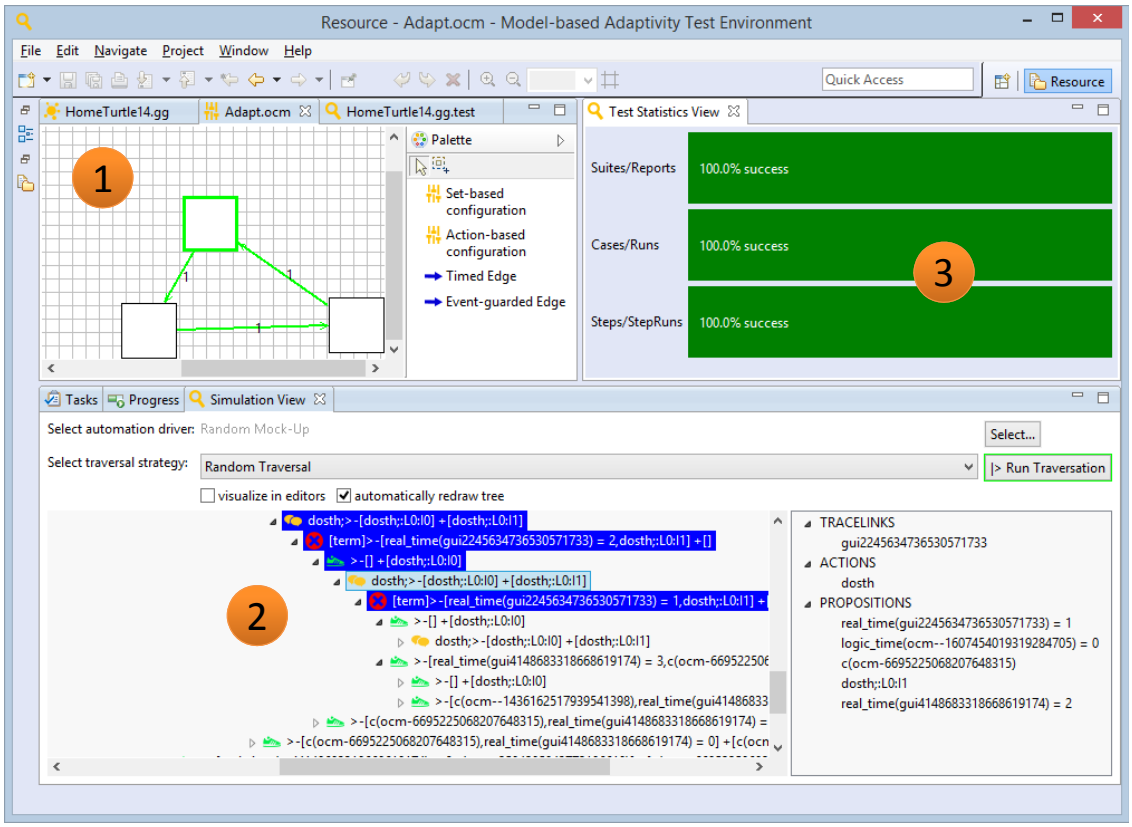


Fig. 10. Screenshot of MATE.

al., SAS leverage both self awareness and context awareness as primitive concepts for higher-level self-adaptive behavior [3]. For context-aware systems, several proposals targeting testing were made as well. The most basic approach in this sense was proposed by Kakousis within the MUSIC project [16]. In order to test a mobile application, a domain specific language was designed that allows for defining timed changes of context information, logging, and evaluation of memory usage. As already discussed, complex systems cannot be covered appropriately with manually defined test cases. However, in the set of inspected works, the MUSIC methodology is the only one that includes means for dealing with time.

Wang et al. discussed in [17] how a context-aware system can be tested by observing how certain context changes trigger adaptation. The basic assumption in their work is that the SUT is based on a context middleware and calls on this middleware's interface are enriched by calls to a dedicated monitoring framework. The points where these calls are made are called context-aware program points (capps). From the monitored execution, a control flow model of capps can be deduced. The context is then manipulated and manipulations are correlated with capps. The resulting information on which context changes trigger which capps can be utilized to generate appropriate test cases. Wang et al.'s method is helpful for stressing the system with good coverage. However, there is no oracle included in the approach, i.e., it cannot be automatically decided whether the triggering of capps was correct. Thus, this approach is less-powerful then our proposal. Another disadvantage is that the developer has to change the SUT's code as the monitoring framework has to be called. In consequence, the approach cannot be considered a pure black box method.

A simulation-based approach was proposed Abeywickrama et al. [18][19]. For SAS design, the authors created a modeling environment SOTA (State of the Affairs) where feedback/control loops can be specified directly in form of hierarchical state charts. In order the examine such systems' correctness, an interactive simulator was added, which visualizes the execution of these models. Thus, the system engineer can observe incorrect states and give manual input where context information is expected. As this approach again does not include any automatic oracle, it can rather be compared to debuggers that execute a program step-wise in order to analyze it manually.

Fleurey et al. showed in [20] how an SAS can be built based on variant models, context variables, and adaptation rules. They also recognized the need for simulation when such systems have to be validated. Thus, they derived a simulation graph and validated it against invariants. In contrast to our black box approach, their validation method is based on design models, which makes it hard for testers to decide whether failures stem from design or implementation.

An advanced strategy was proposed within the DiVA project [21]. The validation of DiVA-based implementations can be performed in two phases: (1) In the early phase, instances of the context model are generated and associated with partial solutions. Those describe how parts of the systems have to be configured after a certain context instance was applied and the corresponding adaptation was performed. (2) In an operational

validation phase, the system's behavior is investigated during a sequence of contextual changes. Sequences are automatically built by a heuristic (so called Multi-Dimensional Coverage Arrays, MDCA). The DiVA validation methods neither consider any system/adaptation interaction, nor do they propose specific test models for constructing an automatic test oracle. Furthermore, in [22], the DiVA authors Munoz and Baudry alternatively propose using statistical models for generating sequences of context changes. They also consider the possibility to define formalized oracles based on direct associations between a change history and an expected adaptation. For a rather complex systems, the management of such associations is much more expensive than with our model (i.e., each possible adaptation has to be modeled separately).

Nehring and Liggesmeyer proposed in [23] a process for testing the reconfiguration of adaptive systems. The validation is performed in six iterations: In the first iteration, a system model is derived and representative workload is prepared by a domain expert and later executed by developers or system engineers. In the second iteration, a system architect checks if structural changes are performed correctly. Thereby, the reconfiguration actions have to be in the correct order such that the system ends in a valid state and the quality of service is only affected minimally during reconfiguration. The third iteration considers data integrity while stressing the system with increasing load. In the fourth iteration step, state transfer between replaced components is investigated. An interaction issue between system transactions and the adaptation is tested in the fifth iteration. The last iteration considers the identity of components and component types before and after adaptation. In comparison to our approach, Nehring and Liggesmeyer assume the adaptive system to be component based and the validation can be sufficiently investigated by a debugger-like tool chain. Thus, their approach is exploratory and hard to use for integration and system testing.

Furthermore, in the area of self-organizing systems, Eberhardinger et al. proposed an approach called *Corridor Enforcing Inrastructure (CEI)* [24]. As MATE, the CEI concepts rely on running a feedback loop that monitors the test object and checks its validity continuously. In contrast to MATE, CEI is more focused on non-functional qualities of the SUT as the definition of validity is based on constraints. The constraints indicate a *corridor of correct behavior (CCB)*, which the SUT is expected never to leave. By extracting *environmental variation scenarios (EVS)* from the SUT's specification, the system is examined in different situations. EVS extraction can also be automated by the use of a model checker that finds sequences of interactions within communication protocols. Eberhardinger et al.'s approach is well-suited for this kind of system and especially for checking non-functional properties of self-organizing systems. In comparison to MATE, it is not yet clear, how the functional complexity of SAS can be tackled and concisely modeled within the CEI concepts.

VIII. Conclusions and Future Work

In this paper, we presented a concept to build black box simulation models for validation of SAS. The models are separated in different components, each matching a certain step inside the proposed Counter Feedback Loop principle. We showed how these interacting model artifacts can be employed for both simulation and test case generation. For this purpose, an enabling infrastructure was outlined. Both test case and simulation execution can be reported in the same manner. Furthermore, we illustrated an excerpt of a simulation run along our example model. The system was implemented and is deployed in our lab.

Our models are based on UML class models, state charts, and equivalence class trees with invariants. Automata communicate by events such that the different concerns of the scenario process and adaptation can be separated. Our approach does not rely on any design model such that engineers are able to build discrete simulation models of arbitrary self-adaptive systems. The methodology comprises a process of classifying environment variability and defining an explicit model on its change. Using this toolset, we match the goals (2)-(5) as stated in Section III. Goal (2)–*Correct adaptation initiation* is considered by letting Adaptation Models receive signal events from the Environment Reconfiguration Models. Thus, the change in context can be causally connected with an adaptation of the system. As Adaptation Models define an operational order of adaptation actions, goal (3)-*Correct adaptation planning* is dealt with. Goal (4)-*Consistent adaptation/system interaction* can be validated as the Process Model accesses the state of the Adaptation Models and defines conditions on this state. Thus, the system's adaptive behavior can be defined. As Adaptation Models can also check an adaptation's outcome by assertions, goal (5)-*Correct adaptation execution* is addressed.

In our future work, we are going to enrich the employed formalism (i.e., state charts, equivalence class trees, etc.) for more compact definitions and experiment with more complex scenarios in order to expand the evaluation. Concerning the improvement of formalism, e.g., we consider using Petri nets as they are more flexible in describing parallelism and synchronization, which is especially important when multiple widely-independent system parts interact.

Furthermore, we considered that it may be beneficial to provide alternative environment reconfiguration model types. While state charts can only model very less-complex and explicitly specified states, data graphs, movement profiles, or event differential formulas could provide a more dynamic representation. For instance, with graphs and differential formulas, data changes can be correlated with discrete simulation time precisely. Instead, changing locations of objects that effect the SUT could be modeled using pre-defined paths that are triggered by simulation time as well.

In summary, we aim at providing a complete test generation and simulation environment that can be employed for almost arbitrary SAS. Our central assumption is that all considerable SAS comply with the MAPE-K loop principle. In order to evaluate this proposition, further case studies will be performed in future work as well. Different scenarios with autonomous systems are considered for this purpose, e.g., SAS-controlled drones and automotive systems.

Acknowledgment

This work is funded within the projects #100084131 and #100098171 (VICCI) by the European Social Fund as well as CRC 912 (HAEC) and the Center for Advancing Electronics Dresden (cfaed) by Deutsche Forschungsgemeinschaft.

REFERENCES

[1] G. Püschel, C. Piechnick, S. Götz, C. Seidl, S. Richly, and U. Assmann, "A black box validation strategy for self-adaptive systems," in Proceedings of The Sixth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE). XPS Press, 2014, pp. 111–116.

[2] B. H. C. Cheng et al., "Software engineering for self-adaptive systems: A research roadmap," in Dagstuhl Seminar 08031 on Software Engineering for Self-Adaptive Systems, 2008, pp. 1–26.

[3] M. Salehie and L. Tahvildari, "Self-adaptive software: Landscape and research challenges," ACM Transactions on Autonomous and Adaptive Systems, vol. 4, no. 2, May 2009, pp. 14:1–14:42.

[4] J. O. Kephart and D. M. Chess, "The vision of autonomic computing," Computer, vol. 36, no. 1, Jan. 2003, pp. 41–50.

[5] IABG, "V-Modell XT 1.4," http://v-modell.iabg.de, visited 04/01/2014, 2012.

[6] M. Utting and B. Legeard, Practical Model-Based Testing: A Tools Approach. Morgan Kaufmann, 2010.

[7] G. Püschel, S. Götz, C. Wilke, and U. Aßmann, "Towards systematic model-based testing of self-adaptive software," in Proceedings of The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE). XPS Press, 2013, pp. 65–70.

[8] "TurtleBot 2," http://turtlebot.com, visited 04/01/2014.

[9] Object Management Group (OMG), "Unified Modeling Language (UML) specification, version 2.4.1," http://www.omg.org/spec/UML/2.4.1/, visited 04/01/2014.

[10] Object Management Group (OMG), "Object Constraint Lanugage (OCL), version 2.3.1."

[11] M. Grochtmann, "Test case design using classification trees," Proceedings of STAR, vol. 94, 1994, pp. 93–117.

[12] G. Püschel, S. Götz, C. Wilke, C. Piechnick, and U. Aßmann, "Testing self-adaptive software: Requirement analysis and solution scheme," International Journal on Advances in Software, ISSN 1942-2628, no. vol. 7, no. 1 & 2, year 2014, 2014, pp. 88–100.

[13] K. C. Kang, S. G. Cohen, J. A. Hess, W. E. Novak, and A. S. Peterson, "Feature-oriented domain analysis (FODA) feasibility study," DTIC Document, Tech. Rep., 1990.

[14] "Eclipse Modeling Framework Project," http://www.eclipse.org/modeling/emf/, visited 04/01/2014.

[15] C. Piechnick, S. Richly, and S. Götz, "Using role-based composition to support unanticipated, dynamic adaptation - smart application grids," in Proceedings of The Fourth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE). XPS Press, 2012, pp. 93–102.

[16] K. Kakousis, N. Paspallis, G. A. Papadopoulos, and P. A. Ruiz, "Testing self-adaptive applications with simulation of context events," Electronic Communications of the EASST, vol. 28, 2010.

[17] Z. Wang, S. Elbaum, and D. S. Rosenblum, "Automated generation of context-aware tests," 29th International Conference on Software Engineering (ICSE), 2007, pp. 406–415.

[18] D. B. Abeywickrama, N. Bicocchi, and F. Zambonelli, "SOTA: Towards a general model for self-adaptive systems," in Enabling Technologies: , IEEE 21st International Workshop on Infrastructure for Collaborative Enterprises (WETICE). IEEE, 2012, pp. 48–53.

[19] D. B. Abeywickrama, N. Hoch, and F. Zambonelli, "SimSOTA: Engineering and simulating feedback loops for self-adaptive systems," in Proceedings of the International C* Conference on Computer Science and Software Engineering, ser. C3S2E '13. New York, NY, USA: ACM, 2013, pp. 67–76.

[20] F. Fleurey, V. Dehlen, N. Bencomo, B. Morin, and J.-M. Jézéquel, "Modeling and Validating Dynamic Adaptation," in Models in Software Engineering. Springer, 2009, pp. 97–108.

[21] A. Maaß, D. Beucho, and A. Solberg, "Adaptation model and validation framework – final version (DiVA deliverable D4.3)," https://sites.google.com/site/divawebsite, visited 02/01/2014, 2010.

[22] F. Munoz and B. Baudry, "Artificial table testing dynamically adaptive systems," CoRR, vol. abs/0903.0914, 2009.

[23] K. Nehring and P. Liggesmeyer, "Testing the reconfiguration of adaptive systems," in Proceedings of The Fifth International Conference on Adaptive and Self-Adaptive Systems and Applications (ADAPTIVE). XPS Press, 2013, pp. 14–19.

[24] B. Eberhardinger, H. Seebach, A. Knapp, and W. Reif, "Towards testing self-organizing, adaptive systems," in Testing Software and Systems. Springer, 2014, pp. 180–185.

The OM4SPACE Activity Service

A semantically well-defined cloud-based event notification middleware

Marc Schaaf*, Irina Astrova†, Arne Koschel‡, and Stella G. Grivas*

*University of Applied Sciences and
Arts Northwestern Switzerland
Olten, Switzerland
Email: {marc.schaaf|stella.gatziu-grivas}@fhnw.ch
†Institute of Cybernetics
Tallinn University of Technology
Tallinn, Estonia
Email: irina@cs.ioc.ee
‡Faculty IV, Department of Computer Science
University of Applied Sciences and Arts Hannover
Hannover, Germany
Email: akoschel@acm.org

***Abstract*—OM4SPACE provides a cloud-based event notification middleware. This middleware delivers a foundation for the development of scalable complex event processing applications. The middleware decouples the event notification from the applications themselves, by encapsulating this functionality into a component called Activity Service. This paper presents the architecture of the Activity Service, its application scenario, its semantic parameters, the implemented prototype of the Activity Service and the preliminary results of the performance evaluation of the prototype. The contribution of the paper is threefold: (1) we identified a new use case for the application scenario; (2) we extended a list of semantic parameters; and (3) we presented an implemented prototype of the Activity Service.**

Keywords–OM4SPACE; Activity Service; Cloud computing; Complex event processing (CEP); Active database management systems (ADBMSs); Smart grids.

I. Introduction

This paper provides an in-depth overview of the Activity Service, including details of its implementation and performance evaluation. The Activity Service was developed as part of the OM4SPACE (Open Mediation for Service-oriented architecture and Peer-to-peer Active Cloud Components) project [1], [2], [3], [4], [5], [6], [7], [8], which was started as a joint project of the University of Applied Sciences and Arts Hannover Germany and the University of Applied Sciences and Arts Northwestern Switzerland.

The idea behind the OM4SPACE project was to merge an event-driven architecture (EDA), a service-oriented architecture (SOA), complex event processing (CEP) and cloud computing together to provide a semantically well-defined cloud-based event notification middleware for decoupled communication between CEP application components on all the layers of a cloud stack, including IaaS (Infrastructure-as-a-Service), PaaS (Platform-as-a-Service) and SaaS (Software-as-a-Service). By decoupled, we mean that events are posted to the middleware without knowing if and how they are processed later.

The remainder of this paper is organized as follows. Section II gives an overview of a possible application scenario for the Activity Service. Section III presents the motivation for the Activity Service. Section IV describes the architecture of the Activity Service; it is followed by a discussion of its semantic foundation (Section V). Section VI gives an overview of transport technologies supported by the Activity Service. Section VII presents the implementation of the Activity Service. Section VIII evaluates the performance of the (implemented) Activity Service. Section IX gives an overview of the related work. Finally, Section X draws the conclusion, whereas Section XI discusses the future work.

II. Application Scenario

The OM4SPACE project defines an application scenario for CEP in a cloud environment. Such a scenario is placed in the domain of smart grids, whose main goal is to reduce peak energy consumption and energy wastage. This should be enabled, by dynamically controlling energy generation and consumption using active components. The application scenario covers the definition of actors, who participate in a smart grid, and event-based communication between them. Also, it details use cases, which constitute the background for defining semantic parameters later.

The Activity Service suits the requirements of smart grids very well because, on the one hand, smart grids provide active components producing and consuming events and, on the other hand, smart grids are targeted towards heterogeneous distributed cloud environments involving diverse transport technologies. Therefore, the application scenario for the Activity Service was settled into the domain of smart grids.

A smart grid is an electricity network that can intelligently integrate the actions of all users connected to it – generators,

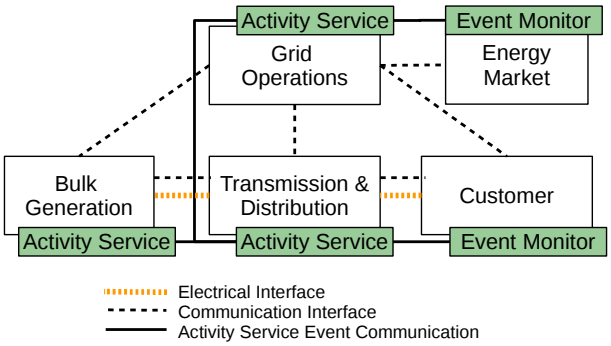


Figure 1. Smart grid actors and relationships among them (adapted from [9]).

consumers and those that do both – in order to efficiently deliver sustainable, economic and secure electricity supplies. Smart grids try to provide all these features with basically three approaches. First, they add dynamics to a power grid. This is done, by replacing passive components with active ones. Second, they add 'intelligence' to the power grid, meaning that the active components are able to communicate with each other and react to changes in the demand of energy. Third, they make the energy-related infrastructure flexible. As a result, new energy sources can easily be added to the power grid, whereas the transmission enables flexible routing.

Today's energy grids are based on mid-20th technologies and concepts. With the lasting gain of energy consumption, the existing infrastructures do not meet future requirements of energy generation, transmission, regulation and availability. Smart grids are 'intelligent' power grids as they are in development by power suppliers and public agencies. With green aspects in mind, smart grids aim to reduce the overall energy consumption and energy wastage.

Basically, there are two parties with interest in smart grids. On the one hand, there are utility providers, such as power authorities. They get enabled to provide stable power grids, get more control over the grids and reduce costs, by avoiding wasting the energy. Providers aim to produce a steady amount of energy, which means no peak or low levels in energy production. Further, the level of energy produced should dynamically fit the current demand of energy. Next to this, the transmission of energy should be minimized. On the other hand, there are customers who can benefit from reliable power grids, energy monitoring and saving, flexible pay scales and convenience features, like remote controlling of electric devices.

In the next subsections, the application scenario will be defined by a set of actors, relationships and use cases.

A. Actors

Actors have certain tasks and interests. In the context of the application scenario, we distinguish four actors: **Bulk Generation**, **Customers**, **Transmission and Distribution**, **Operations** and **Energy Markets** [9].

Bulk Generation simply produces energy. It can be, e.g., generating plants, wind power stations and solar energy plants. In a smart grid, **Bulk Generation** acts as one virtual power plant.

Customers are the consumers of energy. **Customers** may also be part of a virtual power plant as they can generate energy themselves (e.g., using photovoltaic cells). On the **Customers** side, a smart meter is of major importance. Smart meters can be considered as 'intelligent' electricity readers, which are capable of monitoring the energy consumption of attached devices in real time. Furthermore, they provide a history and evaluation of the recorded data. Smart meters are able to control attached and prepared electric devices, e.g., by switching on a washing machine.

Transmission and Distribution is the infrastructure for the grid's power transmission. It is directly connected to **Bulk Generators** and **Customers**.

Operations are a higher controlling and monitoring actor. **Operations** watch the smart grid's state continuously and provide information to the other actors. Furthermore, **Operations** are enabled to actively control the other actors in reaction to certain states of the smart grid.

Energy Markets are commodity markets that deal specifically with the trade and supply of energy (e.g., electricity).

B. Relationships

Basically, in a smart grid, there can be two different kinds of relationships between actors. These are energy transmission and information communication. But the application scenario focuses on the latter only, since this is where events are actually generated. Figure 1 shows all the actors along with the relationships between them. The actor **Operations** represents a smart grid to external partners. Therefore, it is connected to all other actors, thereby enabling to monitor and control the entire smart grid. **Transmission and Distribution** is the connector between **Bulk Generation** and **Customers** for both energy transmission and data transfer.

In the application scenario, two different mechanisms for information communication are defined: events and messages. Events are data that are actively provided by the actors to the smart grid. Events have no dedicated receiver, but are available to other actors as long as they are interested in certain events (this is a simplified view, with no respect to security aspects). Events themselves do not affect the behavior of single actors or smart grids. In fact, events can be collected and processed, which may result in reaction to an event (or a certain set of events). In the application scenario, events are used to add an activity to the actors. With events, actors broadcast information, which indicates their state. Examples of events are the current energy consumption by **Customers**, the current workload of an energy transmission route or the current percentage of 'green' energy produced by **Bulk Generation**.

Messages are data, which are sent from one distinct actor (a sender) to one or more other distinct actors (receivers). Messages are used to directly send commands from one actor to another. The sending of a message may be a reaction to the interpretation of events that currently happened. Examples of messages are **Operations** instructions to **Bulk Generation** to start **Customers** electric devices due to a low energy price or to reduce the energy production due to a lower demand of energy in the smart grid.

In addition to relationships between actors within a single smart grid, the application scenario defines relationships between smart grids. An example of such a relationship is a

situation where one smart grid offers its excessive energy for sale or purchase to another smart grid.

C. Use Cases

Tables I and II illustrate uses cases of various complexity to demonstrate the event-driven behavior of smart grids. A common precondition for all the use cases is that all the actors are completely "smart-grid-ready," meaning that the mandatory infrastructure, such as smart meters and remotely controllable devices, is available.

III. Motivation

Cloud computing is a new paradigm, which quickly finds its way into many IT areas. It offers vast resources and highly dynamic scalability while reducing the required upfront investment costs to a minimum [10]. However, the deployment of CEP applications into the cloud is still a hard task, especially when the applications should benefit from specialized cloud services. This is, however, often required in order to allow for high scalability as for example in case of the event-based communication among the various application components. Many cloud services exist from the various cloud providers as for example Amazon SNS. These services are highly optimized for the cloud as they are providing automatic scaling mechanisms and tight integration with the other provider services. However, their usage typically comes with the price of a vendor lock-in, because they have their own proprietary APIs. As such, they are highly provider-specific.

Current standardization approaches as for example OpenStack are mostly focused on infrastructure aspects and do not cover any standardized support for cloud-based CEP applications. This hampers with interoperability between cloud providers and the outside world because these approaches do not feature well-defined semantics [2], [3].

One simple example from the application scenario, which illustrates the need for well-defined semantics, is an attribute *time* in the following event definition:

```
Event:
    source: s_298
    time: 04:37:21
    temperature: 30
    humidity: 45
```

A time *04:37:21* given in the example above as part of the event definition is problematic, when considered that the time depends on the location where it was made. Thus, a precise time specification needs additional information, such as *according to UTC*. Furthermore, additional information is needed for the correct processing of the time as for example if the given time value represents the point in time when the reported event *occurred* or when it was *detected*. In case where an attribute *temperature* represents some form of an accumulated value, like the average temperature over ten minutes, does the *time* represent the start, the end or some other point in the time during the accumulation period? Does the *time* indicate that the event has already happened or it will happen soon? For example, the *temperature* could have a warning character. If the measuring sensor is approaching a certain threshold, it might send an event upfront to notify about an expected change.

The *time* is only one attribute of an event. But as the example above illustrated, since the semantics of event payload were not explicitly described in the event, further knowledge was required for the correct processing of the event. Unfortunately, cloud services being used to build CEP applications today lack the capability of explicitly specifying such semantics. As a result, when building a CEP application based on those services, the application becomes tightly linked to the provider-specific semantics, which results in a high risk of a vendor lock-in. In many cases, this is also true if measures are taken to hide the service-specific API, because quick abstraction approaches typically do not cover a specification of the overall semantic parameters, which are implicitly provided by the underlying transport technology.

IV. Architecture

Resulting from the motivation and application scenario requirements, we generally aimed at helping application developers to overcome the following challenges when building cloud-based CEP applications:

- Making CEP applications scalable, while minimizing changes to be done to the application design when deploying the applications into the cloud.
- Reducing the risk of vendor lock-in caused by the usage of provider-specific service offerings.
- Dealing with further complexity when crossing the border of a single cloud provider.
- Compensating for lack of semantics.

Within the OM4SPACE project, we developed the Activity Service as the combination of an event notification system and an event processing system to allow for easy event-based communications as well as for event based-rule evaluation and action triggering. In particular, the Activity Service monitors events. After the detection of an event, it notifies the component responsible for executing the corresponding rules (event signaling), which in turn triggers (or fires) these rules into execution. Rule execution incorporates condition evaluation as a first step and, if successful, action execution as the second step. Rules can temporarily be enabled or disabled.

In general, events occur within transactions, whereas rules are executed within transactions. A variety of execution models exist for the coupling of the transactions that raise events, evaluate conditions and execute actions, some giving rise to quite complex behavior. The adaption of such coupling mechanisms will be one of the major challenges in the later phases of the OM4SPACE project.

The general architecture concept follows the unbundling approach introduced in [11]. As such, we divided the Activity Service into three separate components: Event Service, Rule Execution Service and Event Monitor (EM). Each of these components provides one or more well-defined interfaces with a clear definition of their semantics. Thereby the concrete implementation of the different components is interchangeable. The communication between all the components in the architecture is done through events, where an event is any kind of information sent as a notification from one component to another.

Figure 2 illustrates the components of the Activity Service in relation to cloud-based event sources (also called event producers) and event consumers.

TABLE I. Use Case 1 overview

Use Case 1:	Dynamically adjust the bulk energy generation
Actors:	**Bulk Generation**, **Transmission and Distribution**, **Customer**
Preconditions:	*none*
Outcome:	The amount of produced bulk energy is adjusted to the actual energy demand
Trigger:	Calculated changes of the *RouteWorkload* triggered by *EnergyConsumption* events
Description:	To allow for an efficient operation of the whole smart grid, the overall production should be adapted to the actual consumption to reduce overproduction and losses. Modern bulk energy generation is often capable of quickly adjusting its energy production (e.g., gas turbine power plants). Smart meters can provide measurements on their energy consumption patterns allowing for a better estimation of the overall energy need. As such, the measurements generated by the smart meters shall be used to calculate the overall power consumption so that the bulk energy generation can adapt its production schedule.
Procedure:	1. Smart meters installed at the **Customer** premises produce *EnergyConsumption* events based on the accumulated actual energy consumption. 2. The **Transmission and Distribution** provider consumes the *EnergyConsumption* events, calculates the overall route workload for a given time interval and publishes it as an event. 3. The **Bulk Generation** consumes the *RouteWorkload* event and decides if adjustments in its production schedule are required and executes them if needed.

TABLE II. Use Case 2 overview

Use Case 2:	Intelligent energy production / purchasing
Actors:	**Operations**, **Bulk Generation**, **Energy Market**
Preconditions:	*none*
Outcome:	The cost for providing the required energy in the near future is optimized
Trigger:	The energy price on the **Energy Market** falls below a given threshold
Description:	The **Operations** use available energy from the **Energy Market** to optimize its energy cost during periods of high energy consumption.
Procedure:	1. The **Energy Market** raises an *EnergyPrice* event based on the current market situation. 2. The **Operations** receive the *EnergyPrice* event and evaluate it against a given threshold. If the price is below, the **Operations** use the most recent information received from the weather forecast provider as well as the most recent route workload (Use Case 1) to make a prediction for the near future energy consumption and publishes this as an *EnergyConsumtionForecast* event. 3. The **Bulk Generation** consumes the *EnergyConsumptionForecast* event, calculates its own energy production cost for a given time interval and publishes the result as an *EnergyProductionCostEstimate* event. 4. The **Operations** receive the *EnergyProductionCostEstimate* event and correlate the cost with the prices available on the **Energy Market**. If the prices are lower, the **Energy Market** buys the required energy and issues an *ExternalEnergyFeed* event. 5. The **Bulk Generation** consumes the *ExternalEnergyFeed* event and reduces its production schedule accordingly.

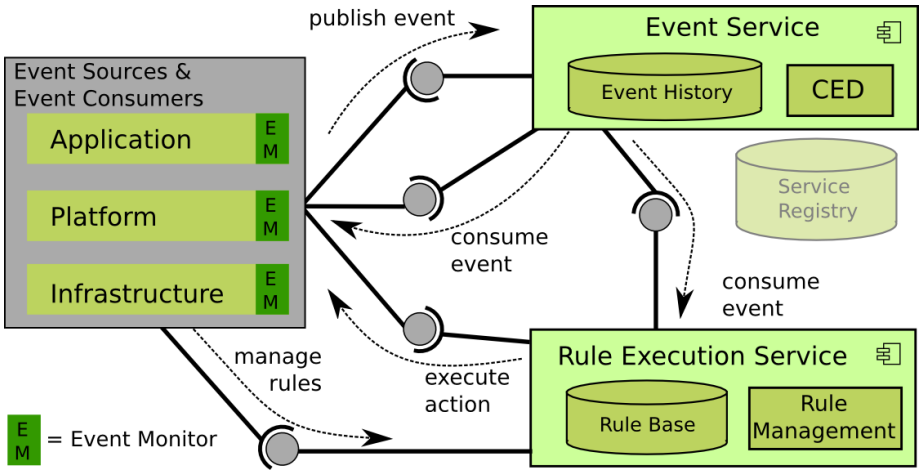


Figure 2. High-level architecture of the Activity Service [8].

A. Event Service

An Event Service represents the central component for the event-based communication. It provides all means necessary to be informed about the occurrence of events from the different event sources and to deliver the events to the subscribed receivers. To receive events from the Event Service, an event consumer has to implement an appropriate Event Handler Service, which needs to be published to the Service Registry. The Event Service discovers those services through the Service Registry and considers them as event subscribers. This mechanism stands in contrast to the commonly used subscriber mechanism where an event consumer needs to know the address of an event source or the event/message broker to directly register the subscription to them.

In the Activity Service, such a direct subscription is not required. Instead we follow the Whiteboard pattern [12] where the Service Registry acts as a 'whiteboard' to allow event consumers to advertise their interest in particular events. We consider this mechanism to be particularly suitable for a dynamic environment, such as the cloud, where the number of instantiated components can change frequently due to automatic scaling activities, making it a hard task to keep all available components up-to-date.

The Event Service further provides the capability to mediate among heterogeneous transport technologies. As part of this mediation process, it interprets semantic parameters to ensure that all participating transport technologies follow the specified behavior, e.g., they all use the requested level of encryption.

In addition to its mediation responsibilities, the Event Service also provides the CEP capabilities. The incoming events are stored into an Event History to support the monitoring of complex events. Its subcomponent Complex Event Detector (CED) evaluates the events and derives new complex events, which are fed back into the processing mechanism so that they can be delivered to registered subscribers or used again as input for the CED. The technical details of the complex event detection process are hidden from the event producers and consumers and can, thus, easily be changed without impacting the rest of the system.

B. Rule Execution Service

A Rule Execution Service extends the CEP capabilities of the Event Service by allowing for more complex rules, which are allowed to access additional background knowledge as part of their processing. The outcome of these extended rules is not required to be a new complex event, but can also represent the execution of an external action, like a remote service invocation. In detail, the rules result in the execution of action handlers. Such an action handler needs to be implemented by each of the components that are to be called from within rules.

The Rule Execution Service receives events from the Event Service to evaluate them against sophisticated rules. Therefore, it acts as an event consumer of the Event Service when registering an appropriate Event Handler Service. The evaluated rules are stored in a Rule Base, which can be managed by a special Rule Management Service.

C. Event Monitors

The Activity Service needs to obtain and process events from heterogeneous event sources that might even be spread across a single cloud. On the one hand, we consider active event sources, such as sources supporting triggers and callback mechanisms typically found in active database management systems (ADBMSs) or sources with internal triggers. On the other hand, we consider passive event sources, such as protocoled sources, which write all their actions into log files. For example, a smart meter usually realizes both types of event sources.

With the requirement to support different types of heterogeneous and highly distributed event sources, we designed a subcomponent called Capsule that hides from the Event Service all the details of a raw event source producing the event payload in a provider-specific format. In particular, the Capsule is responsible for converting the provider-specific format to the format used by the Event Service (e.g., web service calls) and annotating events with semantic parameters.

Figure 3 gives an overview of the architecture of a Capsule. This figure illustrates the raw event source together with the matching Capsule as part of the event producer (each producer has exactly one Capsule). The event producer can reside on any layer of the cloud stack (e.g., IaaS, PaaS and SaaS). Not shown in the figure is the unique capability of the Capsule to utilize event sources outside of the cloud as for example in case of an incoming shipment into a warehouse.

V. Semantic Parameters

The OM4SPACE project aimed at bringing the Activity Service into a cloud environment. Clouds are highly heterogeneous distributed environments, in which multiple different transport technologies can be used. Furthermore, clouds are likely to contain a plenty of heterogeneous event producers and event consumers. Thus, the Activity Service running in a cloud should be able to deal with any kind of events from any kind of event source (both active and passive). It should also provide the CEP capabilities across multiple proprietary and non-proprietary cloud environments.

Beside the issues concerning the transportation of events, the events themselves have to be more meaningful to be processed properly. Multiple different event producers and event consumers may have different requirements on how to interpret the meaning or context of events. Therefore, the events need to be enriched with semantic parameters.

The general concepts of event-based rule evaluation and action triggering have been established in the context of ADBMSs [13]. Therefore, we defined semantic parameters based on those typically found in ADBMSs. In particular, semantic parameters for the Activity Service $ASSP = ESP \cup TSP \cup DSP$ fall into three categories:

1) **Event semantic parameters** (ESP)
   Event semantic parameters describe the interpretation of events and their payload from a non-domain specific perspective. They describe general aspects of an event as for example the exact semantics of a given event timestamp or the lifetime of the event. In the Activity Service, these parameters are heavily influenced by the semantic parameters known from ADBMSs.
2) **Transport semantic parameters** (TSP)
   Transport semantic parameters describe how data are transferred within the Activity Service and enable

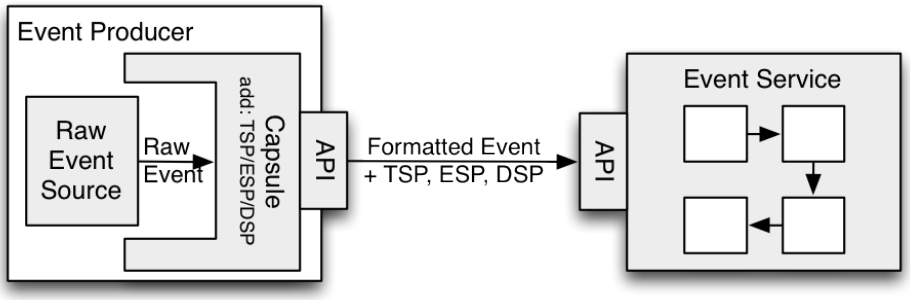


Figure 3. The Capsule annotating events with semantic parameters [3].

TABLE III. The defined event semantic parameters

Signaling point	$sp \in \{pre, post, instead\}$
Granularity	$g \in \{instanceoriented, setoriented\}$
Net effect	$ne \in \{yes, no\}$
Life Span	$ls \in \{explicit, implicit\}$
Consumption policy	$cp \in \{recent, chronicle, continuous, cumulative\}$
Coupling mode	$cp \in \{coupled, decoupled, immediate, deferred\}$
Strategy	$s \in \{parallel, arbitrary, priority, static, dynamic\}$

the usage of heterogeneous transport technologies as they appear in cloud environments due to the various provider-specific services. In other words, transport semantic parameters define in a technology-independent way all the information that must be provided by the underlying transport technology, like the need for confidential event communication or the guarantee that events are delivered exactly once.

3) **Domain-specific parameters** (DSP)
Domain-specific parameters describe the meaning of all relevant information in the application domain that uses the Activity Service. Although being defined separately, these parameters allow for the delivery of domain-specific information as part of the event signaling.

A. Event Semantic Parameters

We took event semantic parameters from ADBMSs and adapted them to the Activity Service. These parameters are defined as $ESP = \{sp, g, ne, ls, cp, cm, s\}$ where each parameter represents the following aspects (Table III):

The **signaling point** describes if an event was raised before the triggering state change happens (*pre*), after the state has already taken place (*post*) or the event replaces the actual state change by giving this notification only (*instead*). We consider all these values as valid options with the change that for the signaling point *pre*, we cannot guarantee that given rules are actually triggered by such an event before the triggering activity has been executed.

The **granularity** indicates the granularity of an event, viz., it can represent a simple singular state change or an aggregation of multiple initial events or state changes, thereby having another granularity. From the perspective of the Activity Service, we support the same granularities as they are typically defined in ADBMSs: *instance-oriented* where each single state change is considered as a single event and *set-oriented* where multiple state changes are considered as one event.

The **net effect** indicates if the event triggering activity had any actual effect and is strongly motivated by transaction handling in ADBMSs. As we are currently not supporting transaction handling, the Activity Service does not handle this parameter yet.

The **life span** defines how long an event is valid for processing. We consider two values (*implicit, explicit*) for the specification of the life span.

The **consumption policy** describes the order how events are processed. In the context of ADBMSs, four policies are defined: *chronicle* where events should be processed by the order of the event creation; *recent* where the last received event should be processed only; *continuous* where the order of receiving is the order of processing (FIFO); and *cumulative* where events are processed as one whole group. We consider the same values but have a special focus on the details of ordered handling as it requires some effort in a distributed system where events are prone to arrive unordered.

The **coupling mode** is also one parameter for transactional behaviors in ADBMSs. It indicates if an event happened within the transaction (*coupled*) or not (*decoupled*). It defines also if an event is thrown immediately or at the end of the transaction (*deferred*). Currently, the Activity Service does not cover transaction handling explicitly and thus, consider only the deferred decoupled value.

The **strategy** defines how the rule execution is triggered if multiple rules would be triggered by an event. The ADBMS semantic definition considers the following values: *parallel* where all matching rules are fired in an unpredictable order; *arbitrary* where one matching rule is picked randomly; *priority* where the rules have priorities and the rule with the highest priority is fired; *static* where a static order is given by an administrator; *dynamic* where the order is generated at runtime. In general, we aim to support all of the available parameters but with one important difference. As for smart grids it is usually the case that multiple rule execution components exist, a global ordering of the rule executions would be hard to achieve. Thus, we consider the given attributes per processing component and not on a global scope. So on a global scope, we only support

TABLE IV. The defined transport semantic parameters

Priority	$p \in \{low, normal, high\}$
Order of delivery	$od \in \{ordered, unordered\}$
Transport reliability	$tr \in \{bestEffort, atLeastOnce, exactlyOnce\}$
Confidentiality	$c \in \{yes, no\}$
Integrity	$i \in \{yes, no\}$
Authentication	$ae \in \{yes, no\}$
Authorization	$ao \in \{yes, no\}$
Transport technology specific	$ts \in \{...\}$

the parallel strategy and allow for a detailed specification per component.

B. Transport Semantic Parameters

The original ADBMS model does not cover distribution. Therefore, to reflect the distributed nature of cloud-based CEP applications, we invented transport semantic parameters. These parameters are defined as $TSP = \{p, od, tr, c, i, ae, ao, ts\}$ where each parameter represents the following aspects (Table IV):

The **priority** allows for the specification of the event importance with regard to its transport. Events with higher priority will be transported more quickly compared to lower priorities if the Activity Service is under heavy load. The current model supports a fixed set of three priority levels *low, normal, high*, which allows for easy mapping to various transport technologies.

The **order of delivery** specifies how events are to be delivered relative to their occurrence. If the ordered delivery is requested, the order in which the events have been published by the corresponding event producer will be kept. The ordering is, however, only guaranteed within the scope of each of the event producers separately. Ordering of the event publications between multiple event producers is not provided. The alternative unordered mode makes no guarantees for the event ordering.

The **transport reliability** parameter enables to specify the need for level of reliability for the event transport level. In general, two categories can be defined: no reliability (*bestEffort*) where no guarantees are given that an event will be correctly transported and guaranteed delivery. The guaranteed delivery is further divided into the categories: *atLeastOnce* where events are guaranteed to be delivered but might be delivered multiple times and *exactlyOnce* where events are guaranteed to be delivered only once.

As cloud-based CEP applications often need to integrate external data sources potentially via an unsecured network like the Internet, the specification of basic security mechanisms is part of transport semantic parameters. In particular, the **confidentiality** parameter enables to specify that events shall be transmitted in such a way that a third party is not able to eavesdrop on them. The **integrity** parameter enables to specify that transmitted events shall be protected from unnoticed modification by a third party. Typically both parameters require some form of authentication and authorization mechanism to be active. The **authentication** parameter enables to specify that an authentication of the communicating parties is required. Based on this, the **authorization** parameter can be used to request that communicating parties are authorized for accessing the transmitted events. As the authorization is only possible once an authentication has been done, it implies that the authentication is active.

In addition to the generally defined parameters, the **transport technology specific** parameter allows for the specification of parameters that are specific to a certain transport technology and thus, understood only by this technology. This parameter can be used, e.g., to optimize a transport technology for lower latency if fast communication is required.

C. Domain-specific Parameters

Domain-specific parameters give the meaning to domain-specific attributes. As an example, consider an event *RouteWorkload* with an attribute *workload* from the application scenario. This attribute has a self-explaining name, which is easy to understand by a human, but not by a software system. Thus, the attribute could be misunderstood by the other components of the system. The attribute needs a domain-specific parameter as for example *measure* with a value of *percent*, thereby explaining that the value of the attribute is given in percent so that all the actors in a smart grid can know the meaning of that attribute.

VI. Transport Technologies

The general communication between the Activity Service components is realized based on the concept of SOA. However, the actual method of message transportation is provided by technology-specific extensions as for example the usage of web services trough an enterprise service bus (ESB) or the use of messaging-based communication trough Amazons SNS or a common message-oriented middleware, like Apache ActiveMQ. This way the Activity Service enables the transparent use of different transport technologies.

The Activity Service is focused on providing a semantically well-defined abstraction from diverse transport technologies to reduce the risk of a vendor lock-in. Consequently, one of the main advantages of the Activity Service is its independence of service providers, such as WebLogic, Amazon and Google. In detail, once an event producer has sent events to the channel, the Activity Service located in the cloud will forward the events to the channel of an event consumer that is subscribed for those events. A decision on which channel to use for sending events is left solely to the event producer. Similarly, a decision on which channel to use for receiving events is left solely to the event consumer. For example, the event producer can select a JMS topic because it is not chargeable, whereas the event consumer can select an SQS queue because it is highly available (i.e., the availability of an SQS queue is not affected if the cloud instance fails). To be used in such scenarios, the Activity Service provides a generalized API along with semantic parameters. The actual transport technology is integrated into the Activity Service as an extension (plug-in), which has the responsibility to map the requested semantic parameters to its technology-specific configuration.

The Event Service as the central component for the event communication is designed to act as a mediator between different transport technologies. This allows the Activity Service to bridge the gap between multiple different provider-specific environments. Due to the explicit definition of the semantic parameters, application developers can rely on the

specified behavior even in complex setups where the event communication needs to be handled with multiple different transport technologies. Figure 4 illustrates such a scenario where event data are received from event producers outside a cloud through two transport technologies: Web Services Eventing (WSE) and web services (WSs). The event data are received by an Event Service, which is also located outside of the cloud that mediates between the aforementioned technologies and JMS-based communications link to another Event Service, which is located in the cloud. In its turn, this second Event Service mediates between JMS and the cloud internal communication service, Amazon SNS/SQS, which is used by the event consumers in the cloud.

In the current version, the Activity Service supports the following transport technologies: JMS and Amazon SNS/SQS.

A. Java Message Service

Java Messaging Service (JMS) provides an API that contains the abstraction of interfaces and classes, which are to be implemented by channel service provider on the basis of a Service Provider Interface (SPI) Adapter. The basic idea behind JMS is that an application can communicate over the JMS API through message-oriented middleware with any other (including non-Java) applications.

We selected JMS as an example of a topic channel from a "native" provider. As a native provider, Apache ActiveMQ was used. JMS was chosen also because the prototype of the Activity Service was implemented in Java.

B. Amazon SNS/SQS

Amazon provides a Simple Queue Service (SQS) and a Simple Notification Service (SNS). SQS is a web service for Amazon Elastic Compute Cloud (Amazon EC2) to decouple applications with message passing. It provides a distributed queue. Messages sent to an SQS queue are stored there until they are received and deleted by the consumer. SNS is another web service that lets endpoints subscribe for a topic and publish messages to that topic. SNS supports different endpoints, including SQS.

There are advantages of coupling SQS with SNS. SQS is a distributed queuing system, where messages are polled by consumers. Polling inherently introduces some latency in the delivery of messages in SQS unlike SNS where messages are immediately pushed to consumers. By coupling SNS with SQS, this latency can be avoided because SNS enables to send messages via an SQS queue to more than one consumer at the same time.

We selected SNS/SQS as an example of a queue channel from a foreign provider. Because of the decision to support SNS/SQS, EC2 was used as the cloud.

C. JMS vs. Amazon SNS/SQS

Table V summarizes our comparison of JMS and SNS/SQS.

JMS is a component of Java Enterprise Edition (JEE), whereas Amazon SNS/SQS abstracts the JEE-specific details of JMS. The main advantage of JMS over SNS/SQS is its independence of a channel service provider. But this also means that there has to be an administrator responsible for setting up the whole infrastructure. On the other hand, the main advantage of SNS/SQS over JMS is the high availability of a channel, which is not affected if a particular Amazon EC2 instance becomes unavailable. Messages waiting in queues for their delivery are stored redundantly on multiple servers and in multiple datacenters. Another big advantage of SNS/SQS over JMS is that there is no limit on the number of messages or the size of a particular queue. One message body can be up to 64 KB of text in any format (default is 8 KB). Large messages can be stored somewhere else reliably (e.g., in Amazon S3) and passed around a reference to the storage location instead of passing the message itself.

However, because of the dependency of Amazon, SNS/SQS is chargeable. There is a free usage tier for up to 100,000 requests per month. Beyond that, Amazon adds $0.01 per 10,000 requests to the bill. In addition, there is a need to pay for the data transfer. Data are transferred free of charge between SNS/SQS and an EC2 instance but within a single region only. Moreover, an SQS queue is distributed. Due to this fact, there is no guarantee that messages are delivered in the same order as they were sent. A sequence number can be added to every message in order to recover the original order of the messages. However, since SNS/SQS saves copies of messages at different servers of the queue, it might happen that in case of a server breakdown, single copies cannot be deleted and are sent twice to the consumer. Therefore, the consumer needs to be implemented in a way that it can handle these redundant messages. On retrieve-message-request, SNS/SQS delivers messages to some of the servers only. This means that it might happen that not all the messages in the queue are delivered or even no messages are delivered at all, if the number of messages is too low. But if the command is executed often enough, all the messages will be delivered step-by-step.

VII. Prototype

Based on the architecture of the Activity Service, we implemented its prototype in Java. The prototype was mostly focused on overcoming the technological gaps between different environments and cloud providers, by providing support for two different transport technologies (viz., JMS and SNS/SQS) and implementing the capability to mediate between them.

For the actual CEP, both the Event Service and the Rule Execution Service utilized the Esper CEP engine. The action handling was implemented based on web service calls against the defined action handler interface. Furthermore, the architecture was extended with a Registration Service, which provides a discovery mechanism, and a Mediation Layer, which provides the flexibility for different transport technologies.

A. Registration Service

To provide the required service discovery functionality for available event producers and consumers as well as the Event Service instances, a Registration Service was implemented that acts as the central service repository (Figure 5). The Registration Service offers its API based on a web service. Each Event Service registers itself via the offered API to announce its presence. Furthermore, all event consumers and producers register themselves via the API to announce either the events they offer or the events they want to receive.

Once the Registration Service has been informed about new event consumers and producers, or about changes in their registrations, it informs the available Event Services about those registrations. The communication with the Event Service

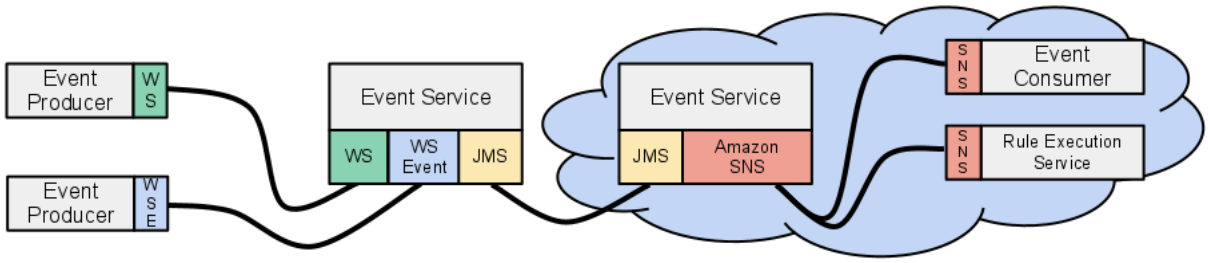


Figure 4. The Activity Service mediating among different transport technologies.

TABLE V. JMS vs. SNS/SQS

Feature	JMS	SNS/SQS
Max queue size	Limit depends on JVM heap and persistence store	Unlimited
Best Quality-of-Service (QoS)	Exactly once	At least once
Channel	Both topics and queues	Queues
Configurable retries	Supported	No
Persistence	Optional	Always
Scalability	Yes, depending of Message Broker	Inherent
Availability	Yes, depending of Message Broker	Inherent
Message order	Supported	Not guaranteed
Auto acknowledge	Supported	No
Message expiry	1 ms to unlimited	1 h to 14 d
Max message size	Unlimited	64 KB (default 8 KB)
Compression / Encryption	Yes	No
Language binding	Java	Java, PHP, Perl and C#

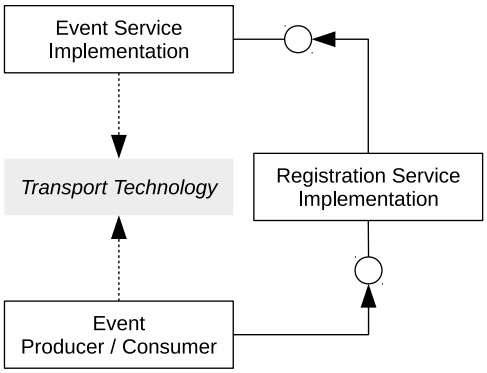


Figure 5. The Registration Service provides service discovery functionality.

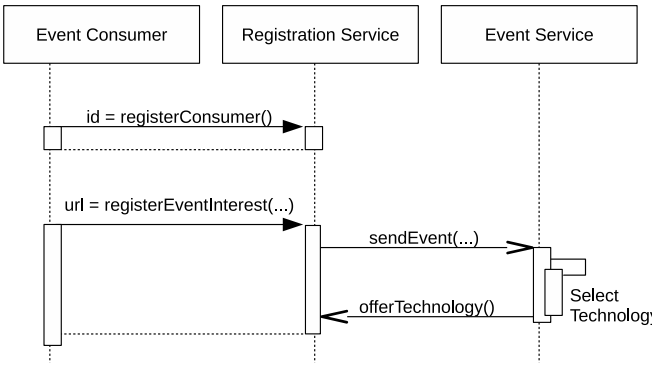


Figure 6. Registration of new event consumer.

is based on special management events. Figure 6 illustrates the registration process of a new event consumer. (The registration process of an event producer is done in a similar way.)

1) A new event consumer first requests a unique id from the Registration Service in order to label all further interactions.
2) Once the id has been assigned, the consumer informs the Registration Service about the type of events it is interested in via a *registerEventInterest()* call. In addition to the *ID* and the *event type*, the call contains a *prioritized list of transport technologies* that are accepted by the consumer for that event subscription.
3) The Registration Service forwards this information to an Event Service that selects suitable transport technologies supported on its side for the requested events.
4) The Event Service then generates the intersection of the list of transport technologies supported by the consumer and the list generated by itself, and selects from the 'common' transport technologies the one with the highest priority.
5) The Event Service then offers a transport technology specific URL, which can be used by the event consumer to receive events from the Event Service. Similarly to the initial call, this information is relayed by the Registration Service to the calling event consumer.

Based on the assigned id, the event consumer could later change or terminate its registration.

In the prototype, the Registration Service has to be started as the first component in order for the Event Service instances to discover and connect to the event consumers and producers.

Also, the Registration Service needs to be implemented as a highly available subsystem, because it is the backbone of the Activity Service dynamic behavior with regard to new or changed event subscriptions.

B. Mediation Layer

One of the main goals of the Activity Service is to bridge the gap between the various communication middleware systems around. Therefore, we designed and implemented a Mediation Layer, which is used by all the components of the prototype and hides the details of the underlying transport technology. In the prototype, the Mediation Layer supports the usage of both JMS and SNS/SQS-based messaging. However, the implementation of additional transport technologies especially from other cloud providers is also planned in the future. To ease such additions, we designed the Mediation Layer as a pluggable system that can easily be extended. Figure 7 gives an overview of the Mediation Layer, which consists of ReceiveMediator and SendMediator.

A ReceiveMediator handles different transport technologies for receiving events from heterogeneous event producers. Therefore, it is expandable for different transport technologies by different plug-ins. In addition, the ReceiveMediator transforms the messages it receives from the event producers to a generic format for the Event Service. This is necessary if the transport technologies keep the messages in different formats in their channels (either topics or queues). As an example, consider the smart meter in a private household that sends its real-time consumption to the distribution network via JMS. The ReceiveMediator receives JMS messages, but the Event Service might know only SNS/SQS because the instance is deployed in Amazon EC2. Therefore, the ReceiveMediator transforms the JMS messages to the XML structures that can then be forwarded to the Event Service via SNS/SQS for further processing.

After the processing, the Event Service sends the (new complex) events to a SendMediator, which distributes the events to heterogeneous event consumers. The SendMediator is a complement of the ReceiveMediator. Like the ReceiveMediator, the SendMediator is expandable for different transport technologies by different plug-ins. In addition, the SendMediator transforms the events it receives from the Event Service to a specific format for the event consumers (e.g., SNS/SQS messages).

The Activity Service does not have to know which of the event producers send events to it. But the Activity Service has to know which of the event consumers want to receive events from it. Therefore, the SendMediator needs an Event Consumer Repository. With such a repository, there is the possibility to store and query information about the event consumers. In the simplest case, this repository could be a database table with one entry for each event consumer. In particular, the Event Consumer Repository has to store the following information:

- The supported transport technologies for each event consumer.
- The values for each transport semantic parameter per event consumer.
- For each event consumer, the event types it is interested in. Not every event consumer wants to receive every kind of events. There are only certain types of events of interest. These types have to be stored in the repository. For example, the distribution network is interested in the real-time consumption of the connected households only.

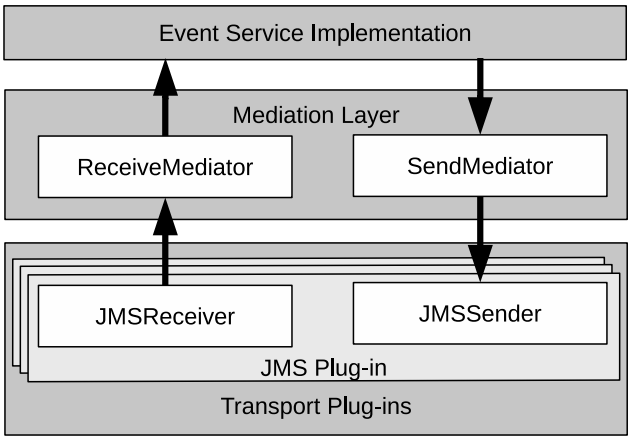


Figure 7. High-level architecture of the Mediation Layer.

With this information, the SendMediator knows if and how it should forward the events to the (registered) event consumers.

For the registration, an event consumer has to send a registration event to the SendMediator. This event must contain information like the used transport technologies and subscribed events. It must also contain the information about the channels of each event consumer so that the SendMediator knows where it has to forward the events later.

C. Capsule

A Capsule acts as the bridge between foreign (i.e., non-Java) applications and the Activity Service, by providing the functionality to forward events from a non-Java application to the Event Service as well as to forward events from the Event Service to the application. As such, the Capsule is implemented as a Java library that can directly be used by the event producing or consuming software. The Capsule also uses the Mediation Layer for supporting multiple transport technologies, while hiding the details from the application that uses the Capsule. The required configuration of the transport technology is possible via a separate XML configuration file and thus, independent of the application:

```
<Event transportType="jms">
  <tsp>
    <key name="choosingPrio">90</key>
    <key name="confidentiality">true</key>
    <key name="integrity">true</key>
    <key name="authorization">true</key>
    ...
  </tsp>
</Event>
```

However, the provided configuration does not contain the values for typical configuration parameters, like the address of the JMS message broker, because these parameters are dependent of the actually used Event Service. Thus, such connection-specific configurations are provided as part of the

registration process, which results in a URL describing the actual endpoint that is to be used (Section VII-A).

In general, the Capsule implementation can support three different modes of operation:

- *Event Consumer Capsule* where the Capsule is used to receive events from one or more Event Services.
- *Active Event Source Capsule* where the event source itself actively notifies the Capsule about new events that shall be transmitted by the Capsule to one or more Event Services.
- *Passive Event Source Capsule* where the Capsule has to detect that new events occurred in the event source, which is not capable to actively notify the Capsule itself.

In the current implementation, we only support the Event Consumer Capsule and the Active Event Source Capsule.

To support the semantic parameters defined by the Activity Service, the Capsule implements the event enrichment process, when it acts as an active event source. In this mode, it annotates each forwarded event with semantic parameters that can be specified in a separate XML configuration file. In particular, the Capsule receives the "raw" events, detects their type and enriches them with the correct values for event semantic and domain-specific parameters. However, not all of the event semantic parameters are set within the Capsule. In particular, event semantic parameters such as for example **consumption policy** and **strategy** cover parts of the rule execution and, thus, require knowledge about the used rules. This knowledge should not be placed into the Capsule because of the maintenance reasons. Therefore, these parameters are set within the Activity Service instead.

Furthermore, the Event Handler Service is responsible for the enrichment of the events with the correct values for transport semantic parameters. This task is not performed by the Capsule either because the transport technologies should be independent of a specific Capsule and, thus, the decisions how to transport the events are done later on within the Event Handler Service.

VIII. Performance Evaluation

As the Activity Service introduces an additional layer of abstraction, we suggested that the Activity Service is likely to have an impact on the overall communication performance. To check if our suggestion is true and to determine if a significant performance impact exists, we measured the Activity Service communication performance and compared the results against measurements taken by direct usage of the underlying transport technology. Each measurement was done with a different number of events (viz., 100, 500 and 1000) to see how the event number affects the performance. Next, we give an overview of these results, which were initially published in [1].

Figure 8 summarizes the test results for the direct usage of a JMS topic (Figure 9.A) and the usage of the Activity Service as a mediator between two JMS topics (Figure 9.B). As expected, the communication via the Activity Service was slower than the direct communication. But JMS still demonstrated a very good performance in all the tests even being interconnected with the Activity Service. For example, sending and receiving 100 events via JMS interconnected with the Activity Service took only 1286 msecs.

We expected that the time would increase with an increase of the number of events. Indeed, for sending and receiving 500 events, JMS interconnected with the Activity Service needed 3184 msecs more than for sending and receiving 100 events. However, of peculiar interest is the fact that for sending and receiving 1000 events, JMS interconnected with the Activity Service needed only 305 msecs more than for sending and receiving 500 events. In both cases, the average time was about 4500 msecs. Therefore, we suggested that extra time needed for sending and receiving 100 events was the time that the Activity Service needed for initialization.

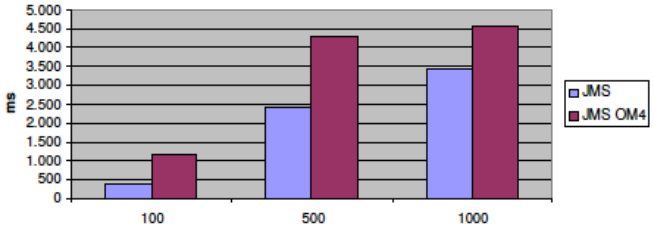


Figure 8. Comparison of event throughput via direct JMS communication and JMS communication through the Activity Service [1].

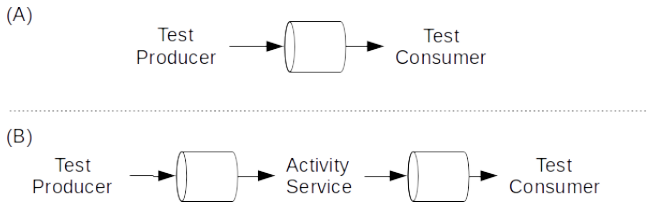


Figure 9. Test set-up for comparison of direct JMS to the Activity Service based JMS communication.

We conducted the same tests with SNS/SQS as the other currently supported transport technology. As expected, due to the distributed nature of an SQS queue, SNS/SQS alone was much slower than JMS alone. Furthermore, due to higher network delays caused by the usage of the cloud service and a less efficient implementation of the event consumer in the prototype, it resulted in a drastic reduction of the relative overhead of the Activity Service.

In general, the test results proved that the additional abstraction layer introduced by the Activity Service also introduces additional performance overhead. This certainly poses a problem for high performance/high throughput application scenarios such as smart grids that need to address the challenges related to the constantly increasing number of events and near real-time reaction on those events. However, the impact becomes less severe once the communication takes place over the across the border of a single cloud or network due to the added latency.

IX. Related Work

The foundation for the Activity Service can be found in the previous work on ADBMSs [14], [15], [16], [17]. An ADBMS provides a rule and execution model with well-defined proven semantics. It also provides a rule definition

language to specify event types, conditions, actions, and their assembly into rules. Much work has been done in the ADBMS context regarding the detection of so called complex events [18], [19]. Complex events are expressions of an event algebra, which are formulated over primitive or complex event types by means of algebraic operators (e.g., say E1 and E2 are event instances, conjunction (E1 ˆ E2) means that E1 and E2 occurred independently of their sequence. A number of technologies for the discovery of complex events are used, such as finite state automates, Petri nets and event trees [14].

In addition to the common database functionality, an ADBMS offers the capability to react to predefined events, by executing the appropriate rules. It provides connectors for event detection, condition evaluation and action execution; all of these are exposed as an overall functionality of an active mechanism. In ADBMSs, the active mechanism is usually closely tied to the systems as a whole. (This is due to the monolithic architecture of ADBMSs.) Therefore, one step beyond the work in ADBMSs go approaches to unbundling the active mechanism from ADBMSs and making it usable in other contexts [11]. For example, in [20], it was prosed to integrate the active mechanism into a rule service for CORBA-based distributed environments. In fact, this work formed a solid starting point for the development of the Activity Service.

Furthermore, we used the original ADBMS model for the definition of semantic parameters for the Activity Service (viz., event semantic parameters). However, the ADBMS set of semantic parameters does not yet cover central aspects that arise from the distributed nature of cloud-based CEP applications. Moreover, they do not cover domain-specific parameters, which can greatly ease the development of cloud-based CEP applications. Therefore, we greatly extended the ADBMS set of semantic parameters with transport semantic parameters and domain-specific parameters for the Activity Service.

Being an important part of the Activity Service, distributed event monitoring systems [19], [21] are an excellent instrument for (distributed) monitoring systems and can contribute to the general monitoring principles of the Activity Service. However, such systems mainly focus on primitive (mostly pure) event sources, like operating system level signals. The Activity Service, on the other hand, has to deal with event sources that are typically found in heterogeneous cloud environments. Event modeling aspects and semantics often lack precision [19] when compared to ADBMSs. Nevertheless, general work on the design of monitoring services for distributed event monitoring systems is valuable for a transfer into the cloud.

Some event monitoring and propagation within the cloud in conjunction with CEP are discussed in [22]. However, rule processing with precisely defined semantics was not the focus there either. On the other hand, several approaches to bringing the CEP technology into the cloud computing domain exist as for example [23], [24]. However, in contrast to the Activity Service, they are application-specific. The first attempt to rule processing within the cloud in conjunction with CEP has been done in [24]. At least some combination of EDA and SOA for the cloud was discussed there. However, the work remains quite high-level and mainly focuses on policy-driven CEP in the cloud. It does not really adapt the active mechanism of an ADBMS to the cloud, in particular, not with well-defined ADBMS semantics.

Web service development standards, such as the business process execution language WSBPEL [25], usually operate on a higher level than the Activity Service. However, they are an excellent example of web services-based systems, which can generate events as for example ”a process or an activity has started or ended.” Such events can then be monitored and acted upon, by using the Activity Service across the whole (heterogeneous) cloud.

X. Conclusion

In smart grids, diverse transport technologies are often involved and a large number of events occur on different layers of the cloud (e.g., IaaS, PaaS and SaaS). This paper provided an in-depth overview of the Activity Service starting from its application scenario, motivation and high-level architecture, and ending with its implementation and performance evaluation. Continuing our successful previous work [1], [2], [3], [4], [5], [6], [7], [8], this paper made additional contributions to the Activity Service. These are a new use case for the application scenario, an extended list of semantic parameters and an implemented prototype.

The Activity Service was developed as a transport technology independent event notification middleware to reduce the risk of a vendor lock-in. It offers an approach to managing events from heterogeneous event sources, processing these events in near real time and triggering appropriate actions on the events. In addition, the Activity Service can be used for monitoring events occurred in the cloud and for scaling CEP applications deployed in the cloud (e.g., starting new virtual machine instances when a certain threshold for the CPU load has been exceeded). A particular highlight of the Activity Service compared to the other work in that area is that the Activity Service is based upon a semantically well-defined rule and execution model. This model is a significant extension of the work originating from the ADBMS area into nowadays distributed, heterogeneous, cloud-based world.

XI. Future Work

In the future, the Activity Service seeks to support more transport technologies, including Web Services Notification, Web Services Eventing and Google App Engine.

A. Web Services Notification and Web Services Eventing

Both Web Services Notification (WSN) and Web Services Eventing (WSE) define a standard web service approach to exchanging notification messages. Both are based on an event-driven or notification-based architecture and use a topic-based publish-subscribe pattern. The difference between the two is that WSN is an OASIS5 standard, whereas WSE is a W3C6 standard. In other words, they are competing specifications with exactly the same idea. However, for the Activity Service, WSN could be a better choice because it supports small devices (with a restricted set of mandatory features) and enables direct and brokered notifications. Also, it offers transformation and aggregation of topics. Furthermore, there are semantic parameters (e.g., available subscription types and broker federations) that are important for high scalability.

B. Google App Engine

Google App Engine's API allows for a persistent connection between a client (HTML or JavaScript) and a server (Python, Java or Go). New information for clients will be pushed through a channel. The clients can subscribe to that channel in order to receive the new information from servers.

A server creates a unique channel for each client and sends a unique token to each client. The server side also receives update messages of the clients and sends those updates to the clients via their channels. A client is primarily responsible for the connection with the channel over the received token. It listens to the channel for updates, makes use of the data and sends the updates to the server. The client id identifies each client on the server. The tokens are responsible for allowing the client to connect and listen to the channel, which is the one-way communication path for the server to send updates to the client.

XII. Acknowledgements

We would like to thank all team members of the OM4SPACE project for their work.

Irina Astrova's work was supported by the Estonian Centre of Excellence in Computer Science (EXCS) funded mainly by the European Regional Development Fund (ERDF). Irina Astrova's work was also supported by the Estonian Ministry of Education and Research target-financed research theme no. 0140007s12.

References

[1] I. Astrova, A. Koschel, A. Olbricht, M., Popp, and M. Schaaf, "Performance evaluation of OM4SPACE's Activity Service," In Proceedings of the 6th International Conferences on Advanced Service Computing, IARIA, pp. 58-61, 2014.

[2] R. Sauter, A. Stratz, S. Grivas, M. Schaaf, and A. Koschel, "Defining events as a foundation of an event notification middleware for the cloud ecosystem," In Proceedings of the 15th International Conference on Knowledge-Based and Intelligent Information and Engineering Systems, LNCS, vol. 6882, pp. 275-284, 2011.

[3] M. Schaaf, A. Koschel, and S. Grivas, "Towards a semantic definition for a cloud-based event notification service," In Proceedings of the 3rd International Conference on Cloud Computing and Services Science, pp. 345-349, 2013.

[4] M. Schaaf, A. Koschel, S. Gatziu, and I. Astrova, "An ADBMS-style Activity Service for cloud environments," In Proceedings of the 1st International Conference on Cloud Computing, GRIDs, and Virtualization, IARIA, pp. 80–85, 2010.

[5] I. Astrova, A. Koschel, S. Grivas, M. Schaaf, I. Hellwich, S. Kasten, N. Vaizovic, and C. Wiens, "Active mechanisms for cloud environments," In Proceedings of the Sixth International Conference on Digital Society, IARIA, pp. 109–114, 2012.

[6] I. Astrova, A. Koschel, L. Renners, T. Rossow, and M. Schaaf, "Integrating structured peer-to-peer networks into OM4SPACE project," In Proceedings of the 27th International Conference on Advanced Information Networking and Applications Workshops, pp. 1211–1216, IEEE, 2013.

[7] M. Schaaf, A. Koschel, and S. Grivas, "Event processing in the cloud environment with well-defined semantics," In Proceedings of the 1st International Conference on Cloud Computing and Services Science, pp. 176–179, 2011.

[8] A. Koschel, A. Hödicke, M. Schaaf, and S. Grivas, "Supporting smart grids with a cloud-enabled Activity Service," In Proceedings of the 27th International Conference on Environmental Informatics for Environmental Protection, Sustainable Development and Risk Management, Berichte aus der Umweltinformatik, pp. 205–213, 2013.

[9] NIST Framework and Roadmap for Smart Grid Interoperability Standards. Release 2.0. NIST Special Publication 1108R2. Available: http://www.nist.gov/smartgrid/upload/NIST_Framework_Release_2-0_corr.pdf Last accessed: November 2014.

[10] M. Armbrust, A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, "Above the clouds: A Berkeley view of cloud computing," Technical report, EECS Department, University of California, Berkeley, 2009.

[11] S. Gatziu, A. Koschel, G. Bültzingsloewen, and H. Fritschi, "Unbundling active functionality," ACM SIGMOD Record, vol. 27, no. 1, ACM, pp. 35-40, 1998.

[12] OSGi Alliance, "The Whiteboard pattern," Technical Whitepaper, Available: http://www.osgi.org/wiki/uploads/Links/whiteboard.pdf Last accessed: November 2014.

[13] D. McCarthy and U. Dayal, "The architecture of an active database management system," In Proceedings of the ACM SIGMOD International Conference on Management of Data, New York, NY, USA: ACM, pp. 215–224, 1989.

[14] K. Dittrich and S. Gatziu, "Aktive Datenbanksysteme, Konzepte und Mechanismen," Int. Thomson Publishing GmbH, Bonn, Albany, Attkirchen, 1996.

[15] N. Paton (editor), "Active rules for databases," Springer, New York. 1999.

[16] The ACT-NET Consortium, "The active database management system manifesto: a rulebase of ADBMS features," In ACM SIGMOD Record, vol. 25, no. 3, ACM, pp. 414-471, 1996.

[17] J. Widom and S. Ceri (editors), "Active database systems: triggers and rules for advanced database processing," Morgan Kaufmann Publishers, Inc., San Francisco, California, U.S.A. 1996.

[18] S. Gatziu and K. Dittrich, "An event definition language for the active object-oriented database system SAMOS," In Proceedings of the Conference on Datenbanksysteme in Büro, Technik und Wissenschaft, Braunschweig, Germany, 1993.

[19] S. Schwiderski, "Monitoring the behaviour of distributed systems," PhD thesis, Selwyn College, University of Cambridge, UK, 1996.

[20] A. Koschel, "Distributed events in active database systems: Letting the genie out of the bottle," In Data Knowledge Engineering, vol. 25, no. 1-2, 1998.

[21] B. Schroeder, "On-line monitoring: a tutorial," IEEE Computer, vol. 28, no. 6, pp. 72–80, 1995.

[22] D. Luckham, "The power of events," Addison-Wesley, Boston, MA, 2002.

[23] G. Wishnie and H. Saiedian, "A complex event routing Infrastructure for distributed systems," In Proceedings of the 33rd Annual IEEE International Computer Software and Applications Conference, pp. 92-95, 2009.

[24] P. Goyal and R. Mikkilineni, "Policy-based event-driven services-oriented architecture for cloud services operation and management," In Proceedings of the IEEE International Conference on Cloud Computing, pp. 135-138, 2009.

[25] OASIS WSBPEL TC. Web Services Business Process Execution Language ver. 2.0, Oasis standard, OASIS, 2007.

Efficient Pattern Application: Validating the Concept of Solution Implementations in Different Domains

Michael Falkenthal, Johanna Barzen, Uwe Breitenbücher, Christoph Fehling, and Frank Leymann
Institute of Architecture of Application Systems
University of Stuttgart
Stuttgart, Germany
{falkenthal, barzen, breitenbuecher, fehling, leymann}@iaas.uni-stuttgart.de

Abstract—**Patterns are a well-known and often used concept applied in various domains. They document proven solutions to recurring problems in a specific context and in a generic way. As a result, patterns are applicable in a multiplicity of specific use cases. However, since the concept of patterns aims at generalization and abstraction of solution knowledge, it is difficult to apply patterns to specific use cases, as the required knowledge about refinement and the manual effort that has to be spent is often immense. Therefore, we introduce the concept of Solution Implementations, which are concrete solution artifacts directly associated with patterns in order to efficiently support elaboration of concrete pattern implementations. In addition, we show how Solution Implementations can be aggregated to solve problems that require the application of multiple patterns at once. We evaluate the presented approach by conducting use cases in the following domains: (i) Cloud Application Architecture, (ii) Cloud Application Management, (iii) Costumes in Films, (iv) User Interaction Design, and (v) Object-Oriented Software Engineering.**

Keywords-pattern languages, solution implementations, pattern application, cloud computing patterns, costume patterns

I. INTRODUCTION

Patterns and pattern languages are well-established concepts in different application areas of computer science and information technology (IT) [1]. Originally introduced to the domain of building architecture [2], the concept of patterns recently got more and more popular in different domains such as education [3], design engineering [4], user interaction design [5], large-scale emergeny management [6], software architecture [7], enterprise application architecture [8], enterprise architecture management [9], cloud application architecture [10], application security [11] or costumes [12]. Patterns are used to document proven solutions to recurring problems in a specific context. However, since the concept of patterns aims at generalization and abstraction, it is often difficult to apply the captured abstracted knowledge to a concrete problem. Thus, pattern application often requires immense manual effort and domain-specific knowledge to refine the abstract, conceptual, and high-level solution description of a pattern to an individual use case. These following examples show that this problem occurs in several domains due to the abstraction of solution knowledge into patterns. For example, if a PHP: Hypertext Preprocessor (PHP) [13] developer uses the patterns by Gamma et al. [14], he or she is faced with the problem that the general solution concepts of the patterns have to be translated to his or her concrete context, i.e., he or she has to implement solutions based on a given programming paradigm predefined by PHP. An enterprise architect who has to integrate complex legacy systems may use the enterprise application architecture patterns by Fowler [8] or the enterprise integration patterns by Hohpe and Wolf [15] to gain insight to proven solutions of his or her problems; but, these are still generic solutions and he or she has to create proper implementations for the systems to integrate. This can lead to huge efforts since he or she also has to consider many constraints given by the running systems and technologies besides paradigms of the used programming languages. A teacher who uses the learning patterns by Iba and Miyake [3] has to adapt them to match his or her prevailing school system with all the teaching methods. To give a final example, a costume designer could use the patterns by Schumm et al. [12] to find clothing conventions for a cowboy in a western film but he or she still has to come up with a specific solution for the specific film.

The above examples show that it is often time consuming to create concrete solutions from patterns, since patterns in general describe proven generic solutions at a conceptual level. To overcome this problem, we suggest that patterns should be linked to the (i) original concrete solutions from which they have been deduced (if available) and (ii) to individual new concrete implementations of the abstractly described solution. Therefore, we introduce the concept of *Solution Implementations* that enables users who want to apply a certain pattern to reuse already existing implementation artifacts for their use cases, which eases the application of patterns and reduces the required manual effort significantly. In addition, our concept supports avoiding errors of manual refinement, since existing solution artifacts can be looked up from patterns.

This paper is an extended version of our former work [1] in which we presented Solution Implementations at the Sixth International Conference on Pervasive Patterns and Applications (PATTERNS 2014). In this article, we now validate the approach of Solution Implementations in detail by conducting additional use cases to show that the concept is domain-agnostic and fundamental in the field of pattern research. The studies covered in this article are conducted in the following domains: (i) Cloud Application Architecture,

(ii) Cloud Application Management, (iii) Costumes in Films, (iv) User Interaction Design, and (v) Object-Oriented Software Engineering.

The remainder of this paper is structured as follows: we clarify the difference between the common concept of pattern solutions and Solution Implementations as separate concrete solution artifacts in Section II. In Section III, we discuss related work and the lack of directly usable concrete solutions in state of the art pattern research. We show how to keep patterns linked to concrete solution knowledge in the form of Solution Implementations and how to select Solution Implementations to establish concrete solution building blocks, which can be aggregated in Section IV. In Section V, we present detailed use cases to show the applicability of the presented concept. We verify the feasibility of the approach by means of implemented prototypes in Section VI and conclude this paper with an outline of future work in Section VII.

II. MOTIVATION

Patterns are human readable artifacts, which combine problem knowledge with generic solution knowledge. Patterns are often organized as pattern languages, i.e., they are related. All patterns of a pattern language follow a canonic pattern format, which is a *template* for documenting all contained patterns. This format typically defines different sections such as “Problem”, “Context”, “Solution”, and “Known Uses”. The problem and context sections describe the problem to be solved in an abstract manner where the solution section describes the general characteristics of the solution in an abstract way. Thus, the general solution is refined for individual problem manifestations and use cases resulting in different concrete solutions every time the pattern is applied. The known uses section is the only place where concrete solutions from which the pattern has been abstracted are described. The description in the known uses section is also only textually but concrete solution artifacts are not related to patterns. Further, the known uses are commonly not extended as the pattern is applied nor do they guide pattern readers during the creation of their own solutions.

Therefore, due to the abstract nature of patterns and generalized issues, most pattern languages only contain some concrete solutions a pattern was derived from in the known uses section. This leads to the problem that the user of the pattern has to design and implement a specific solution based on his individual and concrete use case, i.e., a solution has to be implemented based on the user’s circumstances considering the given pattern. However, many patterns are applied several times to similar use cases. Thus, the effort has to be spent every time for tasks, which were already performed multiple times. For example, the Model-View-Controller (MVC) [16] Design Pattern is an often-used pattern in the domain of user interface design. This pattern was, therefore, implemented for many applications in many programming languages from scratch, as patterns typically provide no directly usable concrete solutions for use cases in a concrete context. Patterns are not linked with a growing list of solutions that can be used as basis to apply them to individual use cases rapidly: each time a pattern should be applied, it has to be refined manually to the current use case. The provided sections such as “Known Uses” and “Examples”, which are part of the pattern structure in most pattern languages [15][17][18], therefore, support the reader in creating new solutions only partially: they provide only partial solution refinements or solution templates as written text but not directly applicable implementations that can be used without additional effort. Thus, the reader of a pattern is faced with the problem of creation and design to elaborate a proper solution based on a given pattern each time when it has to be applied – which results in time-consuming efforts that decrease the efficiency of using patterns.

As of today, patterns are typically created by small groups of experts. By abstracting the problems and solutions into patterns relying on their expertise, these experts determine the content of the patterns. This traditional way of pattern identification, also called the “pattern guru approach” by Reiners et al. [19], creates the two issues already seen: first, the patterns are only hardly verifiable because the concrete solutions they have been abstracted from are mostly not traceable (“pattern provenance”) and second, the patterns document abstracted knowledge, therefore manual effort and specific knowledge is needed to apply them to concrete problems.

Another problem occurs if multiple patterns have to be combined to create a concrete solution. Pattern languages tackle the problem of selecting and applying multiple related patterns to solve overall problems. As shown by Zdun [20], this can be supported by defining relationships between patterns within a pattern language, which assure that connected patterns match together semantically, i.e., that they are composable regarding their solutions. This means that patterns can be used as composable building blocks to create overall solutions. Once patterns are composed to create overall solutions the problem arises that concrete solutions have to be feasible in the context of concrete problem situations. Referring to the former mentioned example of a PHP developer, the overall concrete solution, consisting of the concrete solutions of the composed patterns, has to be elaborated that it complies with the constraints defined by the programming language PHP. So, the complexity of creating concrete solutions from composed patterns increases with the number of aggregated solutions, since integration efforts add to the efforts of elaborating each individual solution. Thus, to summarize the discussion above, we need a means to support the required refinement from a pattern’s abstract solution description to directly applicable concrete solutions and their composition.

III. Related Work

As patterns are human readable artifacts, the template documenting a pattern contains solution sections presenting solution knowledge as ordinary text [2][7][14][18]. This kind of solution representation contains the general principle and core of a solution in an abstract way. Common solution sections of patterns do not reflect concrete solution instances of the pattern. They only provide conceptual sketches of a solution or describe the essence of the solution textually. Thus, they just act like manuals to support a reader at implementing a solution proper for his issues, but they do not provide concrete solution artifacts.

Iterative pattern formulation approaches as shown by Reiners et al. [19][21] and Falkenthal et al. [22] can enable that concrete solution knowledge arising from running projects is used to formulate patterns. Patterns are not just final artifacts but are formulated based on initial ideas in an iterative process to finally reach the status of a pattern. Nevertheless, in these approaches concrete solution knowledge only supports the formulation process of patterns but is not stored in the form of concrete solution artifacts explicitly to get reused when a pattern is applied.

Porter et al. [23] have shown that selecting patterns from a pattern language is a question of temporal ordering of the selected patterns. They show that combinations and aggregations of patterns rely on the order in which the patterns have to be applied. This leads to so called pattern sequences which are partially ordered sets of patterns reflecting the temporal order of pattern application. This approach focuses on combinability of patterns, but not on the combinability of concrete solutions.

Many pattern collections and pattern languages are stored in digital pattern repositories such as presented by Reiners [3], Fehling [24] and van Heesch [25]. Although these repositories support readers in navigating through the patterns they do not link concrete solutions with the patterns. Therefore, readers have to manually recreate concrete solutions each time when they want to apply a pattern.

Zdun [20] shows that pattern languages can be represented as graphs with weighted edges. Patterns are the nodes of the graph and edges are relationships between the patterns. The weights of the edges represent the semantics of the relationships as well as the effects of a pattern on the resulting context of a pattern. These effects are called goals and reflect the influence of a pattern on the quality attributes of software architectures. While this approach helps to select proper pattern sequences from a pattern language it does not enable to find concrete solutions and connect them together.

Demirköprü [26] shows that Hoare logic can be applied to patterns and pattern languages such that patterns are getting enriched by preconditions and postconditions. By considering this conditions, pattern sequences can be connected into aggregates, respectively compositions of patterns where preconditions of the first pattern of the sequence are the preconditions of the aggregate and postconditions of the last pattern in the sequence are accordingly the postconditions of the aggregate. This approach only tackles aggregation of patterns without considering concrete solutions.

Fehling et al. [27][28] show that their structure of cloud computing patterns can be extended to annotate patterns with additional implementation artifacts. Those artifacts can represent instantiations of a pattern on a concrete cloud platform. Considering those annotations, developers can be guided through configurations of runtime environments. Although patterns can be annotated with concrete implementation artifacts, this approach is only described in the domain of cloud computing and must be extended to other domains in order to introduce a means to ease pattern usage and refinement in general.

Mirnig and Tscheligi [29] introduce a general pattern framework based on set theory. This framework provides a general theory of patterns in order to explicate knowledge in pattern structures and relate patterns into pattern languages. Their approach is general due to the definition of patterns and pattern languages by means of set theory and, therefore, provides a domain independent fundamental method to create patterns and pattern languages. Further, they introduce a conceptual mechanism by means of descriptors and targets to combine patterns from different domains, respectively pattern languages. Nevertheless, the approach only deals with abstracted solution knowledge that is captured into patterns and related into pattern languages. Hence, the approach lacks support to deal with concrete solutions. Besides, the approach only describes to combine patterns by means of descriptors and targets in general, but it does not clarify how patterns may work together in concrete use cases. So, the approach does not include a method to resolves functional and non-functional dependencies between patterns to be applied together.

Krleža and Fertalj [30] integrate the concept of patterns into the methodology of model driven architectures (MDA) to assure higher model qualities. They show that patterns can help to purposefully reduce the freedom of modeling in software projects. Patterns are provided for the several abstraction levels of the MDA approach. Further, transformation rules guide users to automatically generate artifacts of more specific levels of the MDA modeling space by considering refinements of a pattern of a more abstract level to a pattern on a more specific level. Thus, relations of patterns in different abstraction levels reduce the number of applicable transformation rules from one level to the other. Further, applicable transformation rules also reduce the number of suitable patterns to be applied on more specific levels, vice versa. So, this design method supports users to build consistent and continuous MDA models covering all abstraction layers. But while patterns and transformation rules are stored to be reused in several use cases, concrete platform specific implementations of patterns are not stored and related to their patterns to be reused directly. The approach also lacks a means to automatically select proper patterns based on criteria, which are defined by a user.

Breitenbücher et al. [31] introduce *Automated Management Idioms* as technology and implementation specific refinements of application management patterns. These idioms can be applied automatically to manage cloud

applications by generating declarative descriptions of the management tasks to be executed. Thus, in general they tackle the same issues as Solution Implementations but only for the domain of application management.

Barzen and Leymann [32] show a formalism to collect concrete solution knowledge in the domain of costumes in films in a structured way to derive costume patterns from the captured concrete solutions. They introduce to use domain specific ontologies to define valid properties and values to describe concrete solutions of the domain. Concrete solutions are classified by means of an equivalence function to mine the essence of a set of concrete solutions. The so captured essence in the form of an equivalence class of concrete solutions makes up a pattern. Further, they generalize the approach that it can be applied also in other domains than costumes in films. Their approach clarifies the correspondence of patterns and concrete solutions and emphasizes the approach presented in this work.

Finally, Fehling et al. [33] show how the approach from Barzen and Leymann [32] can be implemented by means of pattern and solution repositories. Further, they show how patterns and concrete solutions can be interrelated comprehensively across both repositories. This is also a concrete implementation of the approach presented in this paper but only for the domain of costumes in films.

IV. Solution Implementations: Building Blocks for Applying and Aggregating Concrete Solutions of Patterns

In the above section, we summarized the state of the art and identified that (i) concrete solutions are not connected to patterns and that (ii) there are no approaches supporting the aggregation of concrete solutions if multiple patterns have to be applied together. Even though there are approaches to derive patterns from concrete solution knowledge iteratively [21][22], concrete solutions are not stored altogether with the actual patterns nor are they linked to them. Concrete solutions, thus, cannot be retrieved from patterns without the need to work them out manually over and over again for the same kind of use cases. Therefore, we propose an approach that (i) defines concrete, implemented solution knowledge as reusable building blocks, (ii) that links these concrete solutions to patterns, and (iii) enables the composition of concrete solutions.

A. Solution Implementations

We argue that concrete solutions are often lost during the pattern writing process since patterns capture general core solution principles in a technology and implementation-agnostic way. In addition, applications of patterns to form new concrete solutions are not documented in a way that enables reusing the knowledge of refinement. As a result, the details of the concrete solutions are abstracted away and must be worked out again when a pattern has to be applied to similar use cases. Thus, the benefits of patterns in the form of abstractions lead to effort when using them due to the missing information of concrete realizations. We suggest keeping concrete solutions linked to patterns in order to ease pattern application and enable implementing new concrete solutions for similar use cases based on existing, already refined, knowledge. These linked solutions can be, for example, (i) the concrete solutions, which were considered initially to abstract the knowledge into a pattern, (ii) later applications of the pattern to build new concrete solutions, or (iii) concrete solutions that were explicitly developed to ease applying the pattern.

Concrete solutions, which we call *Solution Implementations (SI)*, are building blocks of concrete solution knowledge. Therefore, Solution Implementations describe concrete solution knowledge that can be reused directly. In the domain of software development, Solution Implementations provide code, which can be used directly in the development of an own application. For example, a PHP developer faced with the problem to implement the Model-View-Controller Pattern (MVC pattern) [16] in an application can reuse a Solution Implementation of the MVC pattern written in PHP code. Especially, patterns may provide multiple different Solution Implementations – each optimized for a special context and requirements. So, there could be a specific MVC Solution Implementation for PHP4 and another for PHP5, each one considering the programming concepts of the specific PHP version. Another Solution Implementation could provide a concrete solution of the MVC pattern implemented in Java. Therefore, in this case also a Java developer could reuse a concrete MVC solution to save implementation efforts.

By connecting Solution Implementations to patterns, users do not have to redesign and recreate solutions every time a pattern is applied. The introduced Solution Implementations provide a means to capture existing fine-grained knowledge linked to the abstract knowledge provided by patterns. So, users can look at the connected Solution Implementations once a pattern is selected and reuse them directly. To distinguish between pattern's abstract solutions and Solution Implementations, we point out that the solution section of patterns describes the core solution principles in text format and the Solution Implementations represent the real solution objects – which may be in different formats (often depending on the problem domain), e.g., executable code in software development or real clothes in the domain of costumes. Thus, while patterns are documented commonly in natural text, their Solution Implementations depend mainly on the domain of the pattern language and can occur in various forms. Since many specific Solution Implementations can be linked to a pattern, we need a means to select proper Solution Implementations of the pattern to be applied.

B. Selection of Solution Implementations from Patterns

Once a user selects a pattern, he is faced with the problem to decide which Solution Implementation solves his problem in his context properly. To enable selecting

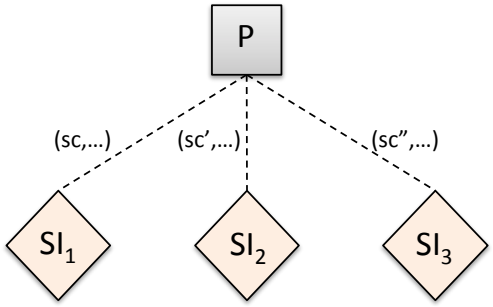


Figure 1. Solution Implementations (SI) connected to a pattern (P) are selectable under consideration of defined Selection Criteria (sc).

proper Solution Implementations of a pattern we introduce *Selection Criteria (sc)*, which determine when to use a certain Solution Implementation. The concept of keeping Solution Implementations linked to the corresponding pattern and supporting the selection of a proper Solution Implementation is shown in Figure 1. Selection Criteria are added to relations between Solution Implementations and patterns. Selection Criteria may be human readable or software interpretable descriptions of when to select a Solution Implementation. They provide a means to guide the selection using additional meta-information not present in the Solution Implementation itself.

To exemplify the concept, we give an example of Solution Implementations from the domain of building architecture. In this domain addressed by Christopher Alexander [2][34], a Solution Implementation would be, for instance, a real entrance of a building or a specific room layout of a real floor, which are described in detail, e.g., by blueprints, and linked to the corresponding pattern [2][34]. To find the most appropriate Solution Implementation for a particular use case, Selection Criteria such as the cost of the architectural Solution Implementation or the used material can be considered. For example, two Solution Implementations for the pattern mentioned above that deals with room layouts might differ in the historical style they are built. Thus, based on such criteria, the refinement of a pattern's abstract solution can be configured by specifying desired requirements and constraints.

To summarize the concept of Solution Implementations it has to be pointed out that solutions in the domain of patterns are abstract descriptions that are agnostic to concrete implementations and written in ordinary text or sketches that illustrate the essential solution principle to support readers. In contrast to this abstract description, we grasp Solution Implementations as concrete solution artifacts, which provide concrete implementation information for particular use cases of a pattern. Solution Implementations are linked to patterns where Selection Criteria are added to the relation between the pattern and the Solution Implementation to guide pattern users during the selection of Solution Implementations.

C. Aggregation of Solution Implementations

The concepts of Solution Implementations and Selection Criteria enable to reuse concrete solutions, which are linked to patterns. But most often problems have to be solved by combining multiple patterns. Therefore, we also need a means to combine Solution Implementations of patterns to solve an overall problem altogether. For this purpose, Solution Implementations connected to patterns can have additional interrelations with other Solution Implementations of other patterns affecting their composability. For example, Solution Implementations in the domain of software development are possibly implemented in different programming languages. Therefore, there may exist various Solution Implementations for one pattern in different programming languages, remembering the above example of the PHP and Java Solution Implementations of the MVC pattern. To be combined, both Solution Implementations often have to be implemented in the same programming language.

This leads to the research question "How to compose Solution Implementations selected from multiple patterns into a composed Solution Implementation?"

Patterns are often stored and organized in digital pattern repositories. These repositories, such as presented by Reiners [3], Fehling [24] and van Heesch [25], support users in searching for relevant patterns and navigating through the whole collection of patterns, respectively a pattern language formed by the relations between patterns. To support navigation through pattern languages, these relations can be formulated at the level of patterns indicating that some patterns can be "combined" into working composite solutions, some patterns are "alternatives", some patterns can only be "applied in the context of" other patterns, etc.

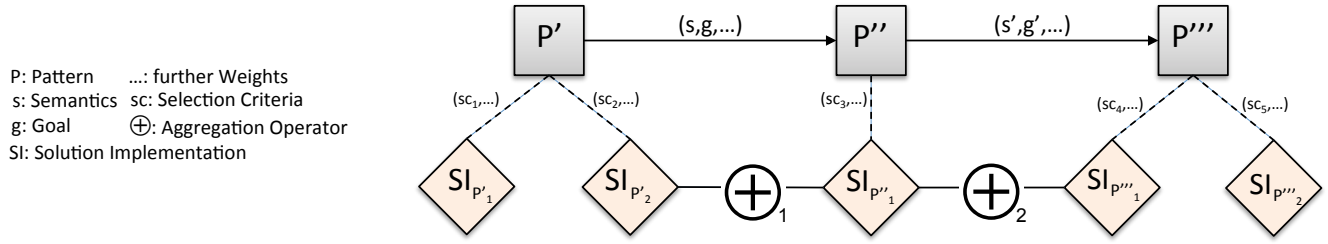


Figure 2. Aggregating Solution Implementations (SI) along the sequence of selected patterns (P).

Zdun [20] has shown that pattern languages can be formalized to enable automated navigation through pattern languages based upon semantic and quality goal constraints reflecting a pattern's effect once it is applied. This also enables combining multiple patterns based on the defined semantics. The approach supports the reader of a pattern language to select proper pattern sequences for solving complex problems that require the application of multiple patterns at once. But, once there are Solution Implementations linked to patterns this leads to the requirement to not only compose patterns but also their concrete Solution Implementations into overall solutions.

We extend the approach of Zdun to solve the problem of selecting appropriate patterns to also select and aggregate appropriate Solution Implementations along the selected sequence of patterns, which is also called solution path.

To assure that Solution Implementations are building blocks composable with each other, we introduce the concept of an *Aggregation Operator,* as depicted in Figure 2. The Aggregation Operator is the connector between several Solution Implementations. It provides the logic to apply two Solution Implementations in combination. Thus, Solution Implementations can just be aggregated if a proper Aggregation Operator implements the necessary adaptations to get two Solution Implementations to work together. Adaptions may be necessary to assure that Solution Implementations match together based on their preconditions and postconditions. Preconditions and postconditions are functional and technical dependencies, which have to be fulfilled for Solution Implementations. In Figure 2, the three patterns P', $P''$and P''' show a sequence of patterns, which can be selected through the approach of Zdun considering semantics (s) of the relations, goals (g) of the patterns and further weights. Solution Implementations are linked with the patterns and can be selected according to the Selection Criteria introduced in the section above. Furthermore, there are two Solution Implementations associated with pattern P' but only Solution Implementation $SI_{P'_2}$ can be aggregated with Solution Implementation $SI_{P''_1}$ of the succeeding pattern P'' due to the Aggregation Operator between those two Solution Implementations. There is no Aggregation Operator implemented for $SI_{P'_1}$, so that it cannot be aggregated with $SI_{P''_1}$, but, nevertheless, it is a working concrete solution of P'. So, in the scenario depicted in Figure 2 an Aggregation Operator has to be available to aggregate $SI_{P'_1}$ and $SI_{P''_1}$.

In general, Aggregation Operators have to be available to compose Solution Implementations for complex problems requiring the application of multiple patterns. Solution Implementations aggregated with such an operator are concrete implementations of the aggregation of the selected patterns. Aggregated Solution Implementations are, therefore, concrete building blocks solving problems addressed by a pattern language.

Aggregation Operators depend on the connected Solution Implementations, i.e., they are context-dependent due to the context of the Solution Implementations. In contrast to the context section of a pattern, which is used together with the problem section to describe the circumstances when a pattern can be applied, the Solution Implementations' context is more specific in terms of the concrete solution. For example, if an Aggregation Operator shall connect two Solution Implementations consisting of concrete PHP code, the Aggregation Operator itself could also be concrete PHP code wrapping functionality from both Solution Implementations. If the Solution Implementations to aggregate are Java class files, e.g., an Aggregation Operator could resolve their dependencies on other class files or libraries and load all dependencies. Afterwards it could configure the components to properly work together and execute them in a Java runtime. In this case an Aggregation Operator is also a runnable program, which implements the logic to combine Java class files automatically. In other domains like building architecture or costumes in films, where Solution Implementations are not concrete programming code but tangible objects, an Aggregation Operator could provide the logic to combine two Solution Implementations by a description of sequential tasks that have to be performed manually.

Thus, an Aggregation Operator composes and adapts multiple Solution Implementations considering their contexts. However, since Solution Implementations of patterns from varying domains are rather different, they have to be aggregated using specific Aggregation Operators. Because different pattern languages deal with different contexts, they can contain different Aggregation Operators to compose Solution Implementations. The validation section will take a closer look at the Aggregation Operators in different domains.

V. VALIDATION WITH PRACTICAL USE CASES

To validate the concept of Solution Implementations, this section conducts detailed use cases focusing on the application of Solution Implementations in the domains of cloud application architecture, cloud management, costumes in films, user interaction design, and software engineering. These use cases show the practical impact of the presented approach by discussing the application of Solution Implementations, Selection Criteria, and Aggregation Operators in the mentioned domains.

A. Use Case 1: Cloud Application Architecture

General Use Case: Business logic is implemented in a component while instances of the component have to be provisioned and decommissioned based on actual workloads. Provisioning and decommissioning shall be managed by another component.

Concrete Scenario: Solution Implementations provide snippets of Amazon Cloud Formation Templates [35],

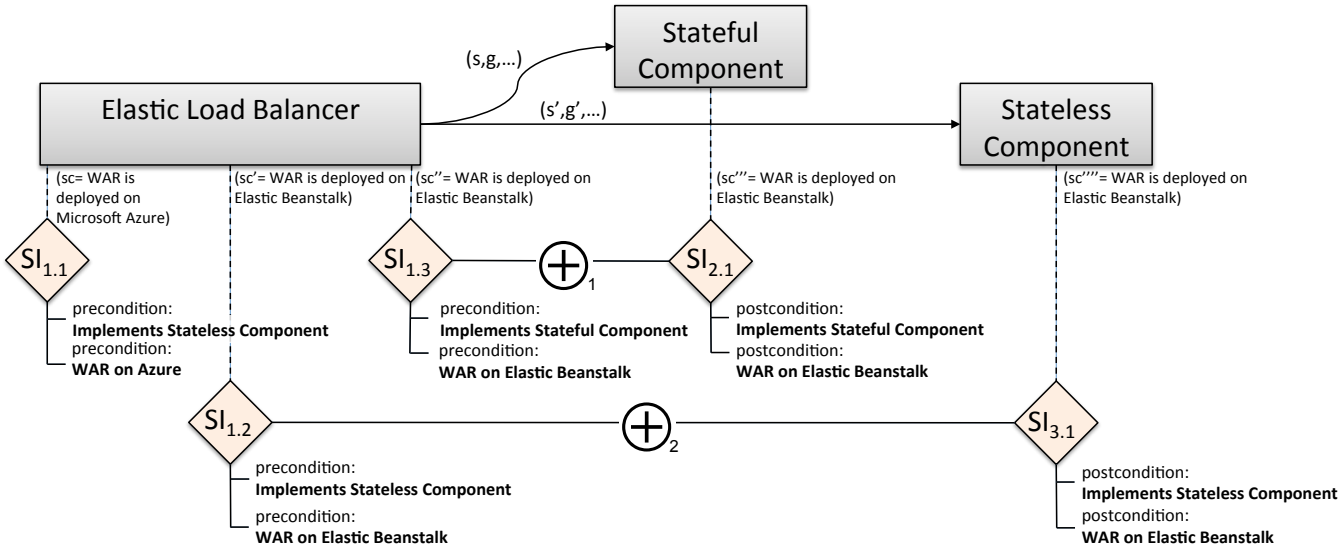


Figure 3. Solution Implementations in the domain of cloud application architecture linked to patterns and aggregated by Aggregation Operators.

which are manipulated by an Aggregation Operator in order to receive a combined configuration file for Amazon's Cloud.

To explain the concept of Solution Implementations in the domain of cloud computing patterns, the example depicted in Figure 3 shows the three patterns *stateless component*, *stateful component*, and *elastic load balancer* from the pattern language and catalogue of Fehling et al. [17][27]. The stateless component and stateful component patterns describe how an application component can handle state information. They both differentiate between session state – the state with the user interaction within the application and application state – the data handled by the application, for example, customer addresses etc. While the stateful component pattern describes how this state can be handled by the component itself and possibly be replicated among multiple component instances, the stateless component pattern describes how state information is kept externally of the component implementation to be provided with each user request or to be handled in other data storage offerings. The elastic load balancer pattern describes how application components can be scaled out, i.e., how performance is increased or decreased through addition or removal of component instances, respectively. Decisions on how many component instances are required are made by monitoring the amount of requests to the managed components. The elastic load balancer pattern is related to both of the other depicted patterns as it conceptually describes how to scale out stateful components and stateless components: while stateless components can be added and removed rather easily, internal state may have to be extracted from stateful components upon removal or synchronized with new instances upon addition.

As depicted in Figure 3, the stateless component and stateful component pattern both provide Solution Implementations, which implement these patterns for Java web applications packaged in the web archive (WAR) format that are hosted on Amazon Elastic Beanstalk [36], which is part of Amazon Web Services (AWS) [37]. In this scenario, both Solution Implementations provide a configuration file that describes the provisioning on a certain platform. This configuration file must be adapted by specifying the actual application files to be deployed. The elastic load balancer has three Solution Implementations realizing the described management functionality for stateful components and stateless components for WAR-based applications on Amazon Elastic Beanstalk and Microsoft Azure [38]. The Selection Criteria "WAR is deployed on Microsoft Azure", respectively "WAR is deployed on Elastic Beanstalk" support the user to choose the proper Solution Implementation. For example, if $SI_{1.2}$ is selected, the user knows that this results in a concrete load balancer in the form of a deployed WAR file on Elastic Beanstalk. Since a load balancer scales components, it needs concrete instances of either stateless component or stateful component to work with. Thus, the user can select a proper Solution Implementation for the components based on his concrete requirements considering the Selection Criteria of the relations between the patterns stateless component and stateful component and their Solution Implementations. To ensure that Solution Implementations are composable, i.e., that they properly work together, they refine and enrich the pattern relationships to formulate preconditions, respectively postconditions on the Solution Implementation layer. The preconditions and postconditions of the elastic load balancer Solution Implementations, therefore, capture which related pattern – stateless component or stateful component – they expect to be implemented by managed

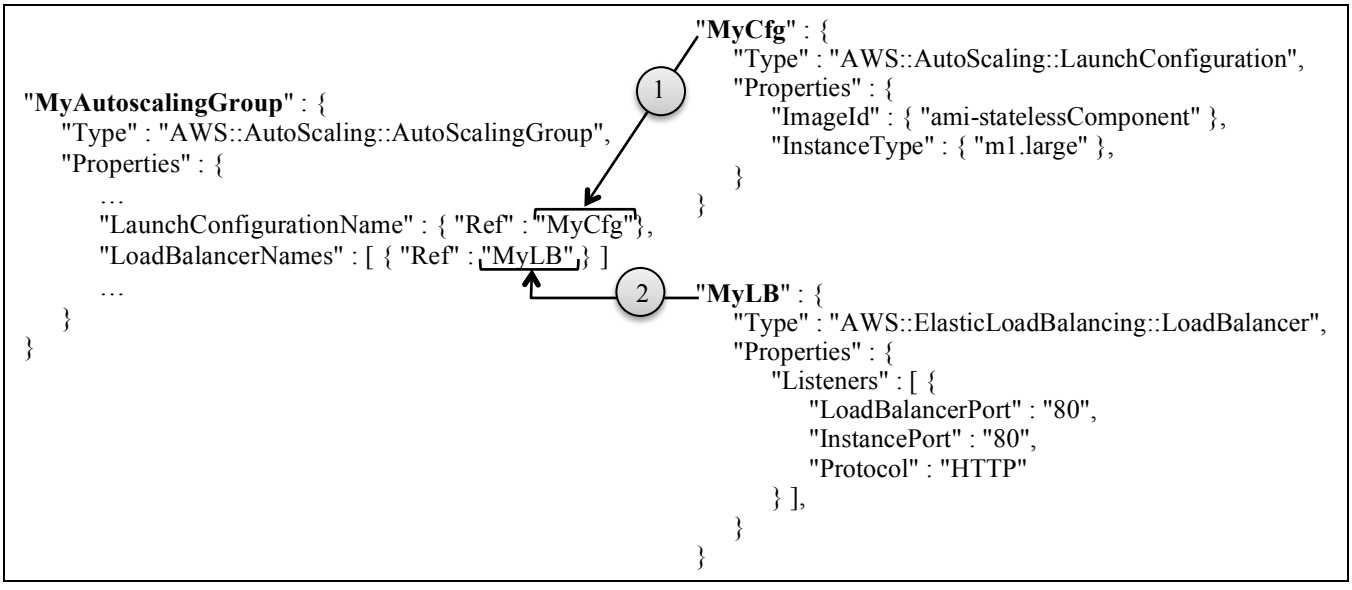
components. Furthermore, they capture the supported deployment package – WAR in this example – and runtime environment for which they have been developed: $SI_{3.1}$ of stateless component has the postcondition "WAR on Elastic Beanstalk" while $SI_{1.2}$ of elastic load balancer is enriched with the precondition "WAR on Elastic Beanstalk" and $SI_{1.1}$ with "WAR on Azure". The previously introduced Aggregation Operator interprets these dependencies and, for example, composes $SI_{3.1}$ and $SI_{1.2}$. During this task, the configuration parameters of the solutions are adjusted by the operator, i.e., the elastic load balancer is configured with the address of the stateless component to be managed. As some of this information may only become known after the deployment of a component, the configuration may also be handled during the deployment.

In the following, this example is concretely demonstrated by an AWS Cloud Formation template [35] generated by the discussed Aggregation Operator. The template is shown in Listing 1. An AWS Cloud Formation template is a configuration file, readable and processable by the AWS Cloud to automatically provision and configure cloud resources. For the sake of simplicity, the depicted template in Listing 1 shows only the relevant parts, which are adapted by the Aggregation Operator. To run the example scenario on AWS, three parts are needed within the AWS Cloud Formation template to reflect the aggregation of $SI_{3.1}$ and $SI_{1.2}$: (i) an elastic load balancer (MyLB), which is able to scale components, (ii) a launch configuration (MyCfg), which provides configuration parameters about an Amazon Machine Image (AMI) containing the implementation of stateless component as well as a runtime to execute the component in the form of an AWS Elastic Compute Cloud (EC2) [39] instance and, (iii) an autoscaling group (MyAutoscalingGroup) to define scaling parameters used by the elastic load balancer and the wiring of the elastic load balancer and the launch configuration.

MyLB defines an AWS elastic load balancer for scaling Hypertext Transfer Protocol (HTTP) requests on port 80. Further, MyCfg defines the AMI ami-statelessComponent in the property ImageId, which is used for provisioning new instances by an elastic load balancer. The autoscaling group MyAutoscalingGroup wires the stateless component instances and the elastic load balancer at the depicted adaption points one and two by means of referencing the property LaunchConfigurationName to MyCfg and LoadBalancerNames to MyLB, respectively. Since all the mentioned properties are in charge of enabling an elastic load balancer instance to automatically scale and load balance instances of components contained in an AMI, an Aggregation Operator can dynamically adapt those properties based on the selected Solution Implementations to be aggregated. So, presuming that ami-statelessComponent contains an implementation of $SI_{3.1}$, an Aggregation Operator can aggregate $SI_{3.1}$ and $SI_{1.2}$ by adapting the mentioned properties at the depicted adaption points and, therefore, provides an executable configuration template for AWS Cloud Formation.

The same principles can be applied to aggregate $SI_{1.3}$ and $SI_{2.1}$ because of their matching preconditions and postconditions. By adapting the ImageId of the LaunchConfiguration to an AMI, which runs an AWS EC2 instance with a deployed stateful component, the Aggregation Operator can aggregate $SI_{1.3}$ and $SI_{2.1}$.

Further, $SI_{1.1}$ has precondition "WAR on Azure" and is, therefore, incompatible with $SI_{2.1}$ and $SI_{3.1}$, i.e., $SI_{1.1}$ cannot be combined with these Solution Implementations due to their preconditions and postconditions. The selection of a Solution Implementation, therefore, may restrict the number



Listing 1. Adaption Points configured by an Aggregation Operator in an extract from an AWS Cloud Formation template to aggregate configuration snippets of elastic load balancer and stateless component.

of matching Solution Implementations of the succeeding pattern since postconditions of the first Solution Implementation have to match with preconditions of the second. This way, the space of concrete solutions is reduced based on the resulting constraints of a selected Solution Implementation. To elaborate a solution to an overall problem described by a sequence of patterns exactly one Solution Implementation has to be selected for each pattern in the sequence considering its selection criteria to match non-functional requirements, as well as postconditions of the former Solution Implementation.

B. Use Case 2: Cloud Application Management

General Use Case: An application component has to be migrated to a cloud environment and downtime is acceptable during the migration. In the cloud environment, the number of component instances shall be automatically increased and decreased considering workloads.

Concrete Scenario: Solution Implementations provide concrete solutions by means of executable workflow snippets, which are combined by an Aggregation Operator. This aggregated solution in the form of a combined workflow snippet automatically deploys the application on Amazon's Cloud offering Elastic Beanstalk and configures the automated scaling.

In this use case, we show how the presented approach can be applied in the domain of cloud application management. Therefore, we describe how applying *management patterns* introduced in [17][40] to cloud applications can be supported by reusing and aggregating predefined Solution Implementations in the form of executable management workflows.

In the domain of cloud application management, applying the concept of patterns is quite difficult as the refinement of a pattern's abstract solution to an executable management workflow for a certain use case is a complex challenge: (i) mapping abstract conceptual solutions to concrete technologies, (ii) handling the technical complexity of integrating different heterogeneous management APIs of different providers and technologies, (iii) ensuring non-functional cloud properties, (iv) and the mainly remote execution of management tasks lead to immense technical complexity and effort when refining a pattern in this domain. The presented approach of Solution Implementations enables to provide completely refined solutions in the form of executable *management workflows* that already consider all these aspects. Thus, if they are linked with the corresponding pattern, they can be selected and executed directly without further adaptations. This reduces the (i) required management knowledge and (ii) manual effort to apply a management pattern significantly. To apply the concept of Solution Implementations to this domain, two issues must be considered: (i) selection and (ii) aggregation of Solution Implementations in the form of management workflows.

To tackle these issues, we employ the concept of *Management Planlets*, which was introduced in our former research on cloud application management automation [41][42]. Management planlets are *generic management building blocks* in the form of workflows that implement management tasks such as installing a web server, updating an operating system, or creating a database backup.

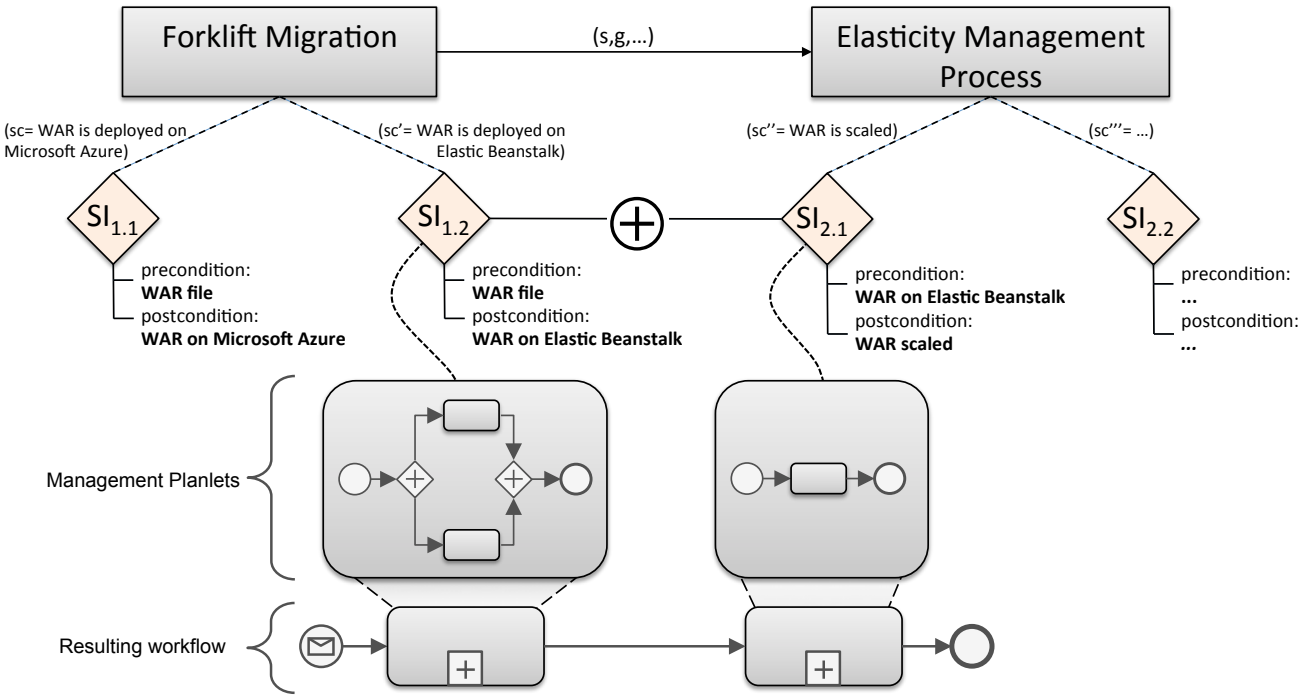


Figure 4. Management Planlets are Solution Implementations in the domain of cloud management linked to patterns and aggregated by an Aggregation Operator.

Each planlet exposes its functionality through a formal specification of its *effects* on components, i.e., its *postconditions*, and defines optional *preconditions* that must be fulfilled to execute the planlet. Therefore, each specific precondition of a planlet must be fulfilled by postconditions of other planlets. Thus, planlets can be combined to implement a more sophisticated management task, such as migrating an application or its components. If two or more planlets are combined, the result is a *Composite Management Planlet (CMP)*, which can be recursively combined with other planlets again: the CMP inherits all postconditions of the orchestrated planlets and exposes all their preconditions, which are not fulfilled already by the composed planlets. Thus, management planlets provide a recursive aggregation model to implement management workflows. Based on these characteristics, Planlets are ideally suited to implement management patterns in the form of concrete Solution Implementations. We create Solution Implementations that implement a pattern's refinement for a certain use case by orchestrating several Planlets to an overall Composite Management Planlet. This CMP implements the required functionality in a modular fashion as depicted in Figure 4.

As stated above, selection and aggregation of Solution Implementations must be considered, the latter if multiple patterns are applied together. For example, Figure 4 shows two management patterns: (i) *forklift migration* [40] – application functionality is migrated with allowing downtime and (ii) *elasticity management process* [17] – application functionality is scaled based on experienced workload. Both patterns are linked to two Solution Implementations, each in the form of Composite Management Planlets that implement the corresponding management logic as executable workflows. The forklift migration pattern provides two Solution Implementations: one migrates a Java-based web application (packaged as WAR file) to Microsoft Azure [38], another to Amazon Elastic Beanstalk [35]. Thus, if the user selects this pattern and chooses the Selection Criteria defining that a WAR application shall be migrated to Elastic Beanstalk, $SI_{1.2}$ is selected. Whether this Solution Implementation is applicable at all depends on the context: if the application to be migrated is a WAR application, then the Solution Implementation is appropriate and the associated Planlet migrates the WAR application to Beanstalk. Equally to this pattern, the elasticity management process pattern shown in Figure 4 provides two Solution Implementations: one provides executable workflow logic for scaling a WAR application on Elastic Beanstalk ($SI_{2.1}$). In this scenario, the workflow simply configures the automated scaling feature, which is natively supported by Amazon Beanstalk. Thus, if these two patterns are applied together, the selection of $SI_{1.2}$ restricts the possible Solution Implementations of the second pattern, as only $SI_{2.1}$ is applicable (its preconditions match the postconditions of $SI_{1.2}$). As a result, the selection of appropriate Solution Implementations can be reduced to the problem of (i) matching Selection Criteria to postconditions of Solution Implementations and (ii) matching preconditions and postconditions of different Solution Implementations to be combined.

After Solution Implementations of different patterns have been selected, the second issue of aggregation has to be tackled to combine multiple Solution Implementations in the form of workflows into an overall management workflow that incorporates all functionalities. Therefore, we implemented a single Aggregation Operator for this pattern language as described in the following: to combine multiple Solution Implementations, the operator integrates the corresponding workflows as subworkflows [43]. The control flow, which defines the order of the Solution Implementations, i.e., the subworkflows, is determined based on the patterns' solution path depicted in Figure 2. So in general, if a pattern is applied before another pattern, also their corresponding Solution Implementations are applied in this order.

C. Use Case 3: Costumes in Films

General Use Case: An actor or an actress playing the role of a superhero that hides his strength by means of boring clothes in his daily live has to be dressed with several costumes. The superhero needs the ability to easily exchange his every day clothes with the superhero costume.

Concrete Scenario: Solution Implementations are provided by means of concrete costumes, which are manually aggregated into one costume.

In the domain of costumes in films, costume patterns can be defined as a proven solution to the design problem for communicating a certain character such as a sheriff or an outlaw by their clothes [12]. A costume transports a lot of information about a character like character traits, moods and social standing, as well as information on the setting of the film. Costume patterns capture the convention of this communication. Like in the other domains, when working with the costume patterns the costume designer needs to spent significant effort to implement the abstract solution description provided by the pattern for a concrete context. When starting to search for the right costumes needed for a certain film, the patterns are of great help by providing the essence of the convention on how to dress characters like the typical superhero or a shy guy in means of being understood and recognized easily by the spectators. For example, the superhero costume probably contains items of clothes like a cape, tight-fitting pants, and a shirt that emphasize the muscles and allow free movements together with a unique logo of this hero. The shy guy, on the other hand, is mostly communicated by a costume of rather pale colors and is dressed in a slightly too big modest suit hiding his face behind big glasses. As this solution is rather abstract, it needs refinement when being applied.

Therefore, in our approach, we suggest the concept of Solution Implementations for connecting the patterns with concrete solutions, meaning descriptions of concrete costumes occurring in films. Since the real tangible costumes are hardly ever kept and stored after the production of a film and since the communicative effect of a costume is retained in films the Solution Implementations are the costumes seen on screen. The descriptions to capture the Solution

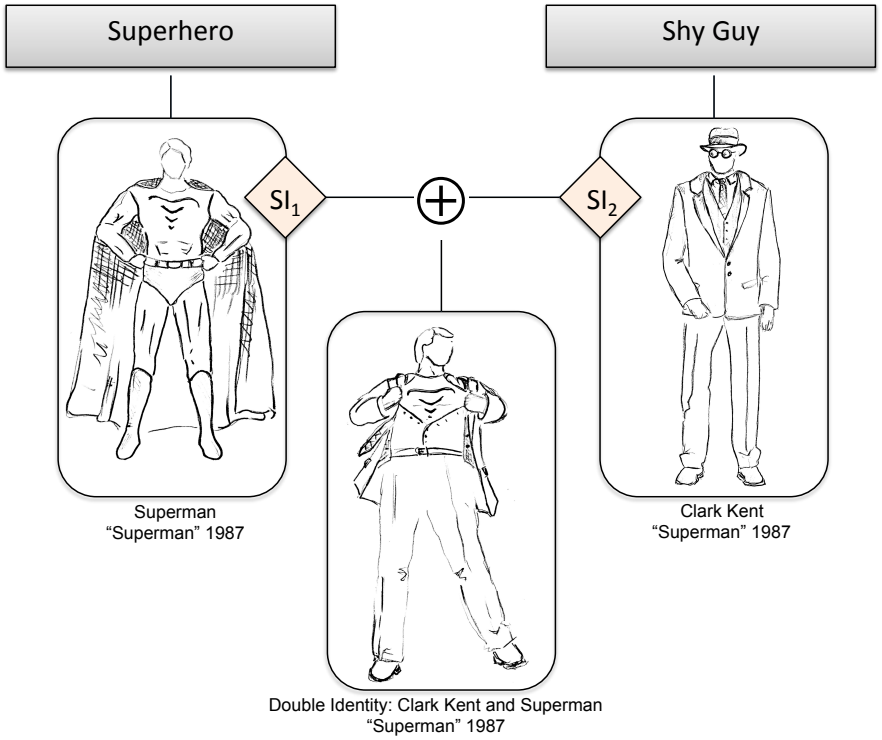


Figure 5. Concrete costumes occurring in the film "Superman" (1987) as Solution Implementations (SI_1, SI_2) of the costume patterns Superhero and Shy Guy are aggregated by an Aggregation Operator.

Implementations contains detailed information on the items of clothes, their material and color, a collection of pictures of the costume as well as contextual information like character traits of the role or its stereotype [32]. Such Solution Implementations can be stored in a Solution Implementation repository [33].

Figure 5 illustrates how the *superhero* pattern, for example, can be connected to the concrete Solution Implementation that the character "Superman" wears in the movie "Superman" (Director: Richard Donner, 1978) or how the *shy guy* pattern can point to the costumes of the character "Clark Kent" in the same movie. But next to the Solution Implementation of the Superman costume, various other Solution Implementations could be connected to the pattern "Superhero" like the Batman or Spiderman costume. Since every pattern can be connected to various Solution Implementations, it is necessary to select the suitable Solution Implementation for the right context. To support finding the right Solution Implementation, the introduced concepts of Selection Criteria as well as defining the pre- and postcondition of the Solution Implementation is also adaptive in the domain of costumes. To find suitable Solution Implementations, i.e., concrete costumes for a concrete film, the Selection Criteria as well as the defined pre- and postcondition of the Solution Implementation can ensure that the costume makes sense in a certain scene. For example, if the Shy Guy pattern shall be applied for Clark Kent in a cold winter scene, other costumes must be taken than if the pattern has to be applied for a scene in summer.

While the concepts of the Solution Implementations, the Selection Criteria, and defined pre- and postconditions are very promising in the domain of costumes, the concept of Aggregation Operators is not always needed: when using a costume pattern to find the right costume, the application of this pattern usually needs just one Solution Implementation and in difference to fragments of code, they are mostly connected together by the storyline and only seldom in a physical way. Nonetheless, there are some situations were *physical* Aggregation Operators are needed. For example, when multiple costume patterns are applied together to one character at once, the corresponding Solution Implementations also need an aggregation and, therefore, need a physical Aggregation Operator. Figure 5 depicts how in the film "Superman" the Solution Implementations of the superhero pattern (SI_1: Superman) and a Solution Implementation of the shy guy pattern (SI_2: Clark Kent) are aggregated together using the Aggregation Operator to build a costume that contains both characters and allows the transformations from one to the other (we omitted Selection Criteria for the sake of simplicity). The necessary adaptations to get those two Solution Implementations to work together would contain actions like making sure that the costume on top needs to be a bit bigger to hide the other, where to store the cape so it is not seen, and how to modify the suite so it does not get torn when being ripped off, for example. We also point out, that in this case, the concept of the Aggregation Operator cannot be automated because the adaption of the costumes in order to fit together has to be done manually by a costume designer.

D. Use Case 4: User Interaction Design

General Use Case: Users need the ability to sign up for accounts of a website. Thus, the users need to provide a password and the sign up process shall only start if the strength of the entered password is validated as strong enough. If a user enters a weak password, he has to be notified that the password needs to be improved.

Concrete Scenario: Solution Implementations provide concrete HyperText Markup Language (HTML) and JavaScript snippets used for designing user interfaces. The final user interface is constructed by aggregating a sequence of Solution Implementations by manipulating associated HTML code.

Patterns are a well-known concept in the domain of user interaction design. A broad number of publications exist that introduce patterns for good user interface designs and user interaction concepts [5][44 - 47]. This use case shows how the approach of Solution Implementations is applied in this domain and, especially, how Solution Implementations from a series of four patterns are aggregated into one combined concrete solution.

As designing user dialogs on websites is a very common issue, many patterns are published that deal with the problem how to design and arrange control elements on a website. Nevertheless, it is still a time consuming effort for a web designer to implement the solution concepts provided by patterns – especially if the concrete website needs to combine several patterns in order to design a complex web interface for users. This is due to the manifold of possible concrete solutions because of the vast number of available technologies to implement websites and control structures. To mention some common technologies today, there are PHP [13], HTML [48], JavaScript [49], Java Servlets [50], JavaServer Pages [51], JavaServer Faces [52], Angular.js [53], jQuery [54], Spring [55], Ruby on Rails [56], Google Web Toolkit [57] and many more. Although websites are rendered using HTML, the different technologies often employ specific concepts to implement a user interface. Unfortunately, this is mostly not plain HTML but a complex combination of server side logic and JavaScript libraries on the browser. In addition, some technologies employ technology-specific constructs and domain-specific languages on server side to specify the control elements of a user interface, which is then transformed into HTML code and the corresponding JavaScript libraries. This means that a developer has to be familiar with language-specific constructs and concepts, complex libraries, and how to combine them in order to refine a pattern's conceptual solution to a concrete implementation. As a result, implementations have to be redeveloped for every technology and use case leading to huge manual efforts.

In the following, we investigate this statement in more detail and assume that a web designer has to implement a website where users can sign up an account by entering a user name and a password. The sign up process shall only start if a safe password is entered (for the sake of simplicity we omit the second password field, which is usually provided for reentering the password to ensure that a user keys in the right password). Therefore, the website has to indicate the strength of the currently entered password. Further, if the user tries to sign up with a weak password, the website should notice him or her about the necessity of a stronger password. This is a very common use case since almost every web shop in the World Wide Web provides such functionality in order to store user specific configurations of the site or the user data for delivery, payment, and invoicing.

In order to realize a website to create accounts, a web designer can use user interaction patterns from [5]. Patterns that are appropriate for the mentioned use case are depicted in Figure 6: *registration, password field, password strength meter* and, finally, *input error message*. The registration pattern describes that a registration form needs control elements to input a user name and a password as well as a button to submit the sign up request. The password field pattern describes that input fields for passwords should not show the password in plaintext. Nevertheless, they should

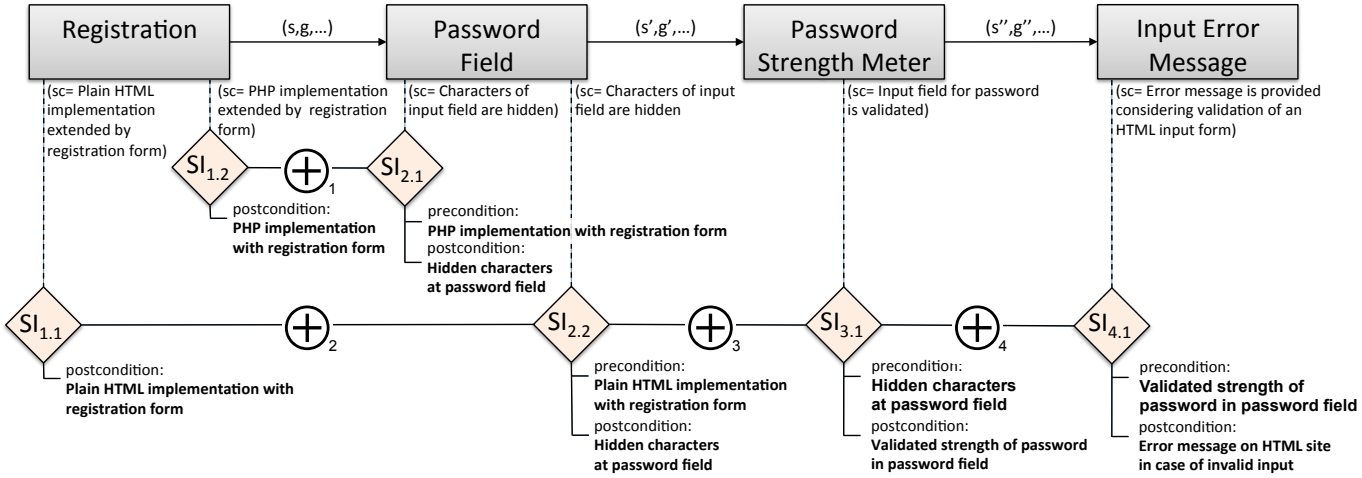


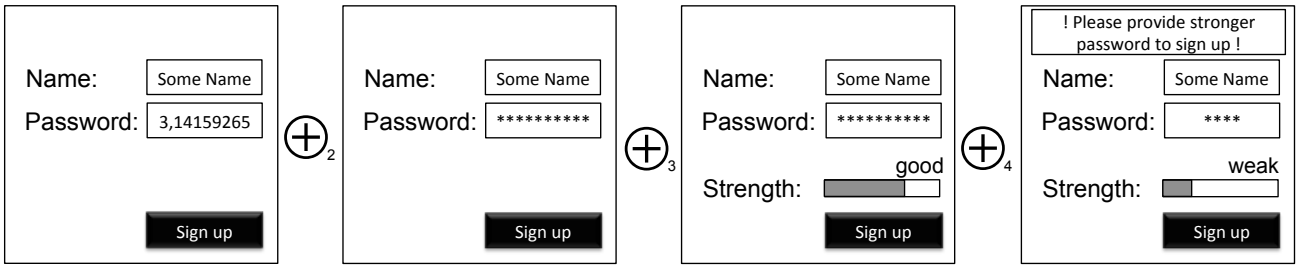
Figure 6. Solution Path of four User Interaction Patterns with related Solution Implementations, which are aggregated by Aggregation Operators.

indicate to the user how many characters have been entered. Further, the password strength meter pattern describes how the strength of a password, i.e., if it is secure or not, can be validated and how a user can be notified about the strength. Finally, the input error message pattern provides a solution how to notify a user about invalid inputs in input fields. It also defines that the website should inform which input field contains the invalid data.

In order to create a concrete solution based on the selected patterns, Solution Implementations have to be selected from all patterns of the solution path. The registration pattern and the password field pattern provide Solution Implementations that extend a plain HTML website ($SI_{1.1}$) or a website coded in PHP ($SI_{1.2}$) as indicated by the Selection Criteria "Plain HTML implementation extended by registration form" and "PHP implementation extended by registration form", respectively. Since the password field shall be protected to avoid unintended discoveries of entered passwords by viewers, either $SI_{2.1}$ or $SI_{2.2}$ have to be combined with $SI_{1.1}$ or $SI_{1.2}$. This is possible since pre- and postconditions of both pairs of Solution Implementations match and Aggregation Operator 1 exists to combine $SI_{1.1}$ with $SI_{2.2}$ as well as Aggregation Operator 2 for $SI_{1.2}$ and $SI_{2.1}$. Since for the following patterns of the solution path – password strength meter and input error message – no Solution Implementations are available in the example depicted in Figure 6, which can be combined with the PHP alternative of password field, we assume that $SI_{1.1}$ and $SI_{2.2}$ are selected. Therefore, $SI_{3.1}$ and $SI_{4.1}$ are also selected because also Aggregation Operators exist to combine them with the previous Solution Implementations along the solution path.

In order to investigate how the Aggregation Operators manipulate the plain HTML file, all aggregations along the solution path are depicted in Figure 7 from left to right. On the left side of this figure, the user interface is illustrated as provided by $SI_{1.1}$. The user interface contains two input fields with their labels "Name" and "Password" as well as a button to submit the sign up request. The password in the second input field still shows the entered characters in plain text. Beneath the sketched user interface, an excerpt of the HTML code provided by the Solution Implementation is shown. The bold letters indicate the registration form with its control elements. After Aggregation Operator 2 has combined $SI_{1.1}$ and $SI_{2.2}$, the input field for the password is manipulated to hide entered characters and only show how many characters are keyed in by means of stars. In plain HTML, this can be achieved by changing the type of the input field from text to password as depicted in the second code snippet in bold letters. Thus, the Aggregation Operator configures the type of the existing input field.

The password strength meter provided by $SI_{3.1}$ extends the HTML file by validation logic implemented in an additional JavaScript file. Besides the logic to determine if an entered password is secure or not, the JavaScript file also contains code to display the strength meter by means of a bar and a label. The more the bar is filled, the more secure the entered password is. To integrate this functionality, Aggregation Operator 3 manipulates the HTML file so that the JavaScript file is loaded, as depicted with the top bold letters in the third code snippet from left. The bottom bold letters in this code snippet shows that the password strength meter is placed between the password field and the submit button as illustrated in the sketch upon the code snippet. To wire the password strength meter with the password input field, the Aggregation Operator has to be configured in order to parameterize the password strength meter with the id of the password input field. The resulting HTML file can be modified by the web designer manually, if the position of the password strength meter does not suit the needs of the




```
<html>
...
<form action="reg.html">
<input id="name" type="text">
<input id="pw" type="text">
<input type="submit"
         value="Sign up">
</form>
...
</html>
```

```
<html>
...
<form action="reg.html">
<input id="name" type="text">
<input id="pw" type="password">
<input type="submit"
          value="Sign up">
</form>
...
</html>
```

```
<html>
...
<script src="strengthMtr.js">
...
<form action="reg.html">
<input id="name" type="text">
<input id="pw" type="password">
<strengthMeter validate="pw">
<input type="submit"
         value="Sign up">
</form>
...
</html>
```

```
<html>
...
<script src="strengthMtr.js">
<script src="inputErrMsg.js">
...
<form action="reg.html">
<inputErrMessage>
<input id="name" type="text">
<input id="pw" type="password">
<strengthMeter validate="pw">
<input type="submit"
          value="Sign up">
</form>
...
</html>
```

Figure 7. Aggregation Operators combine Solution Implementations by adapting HTML code.

website structure etc.

Finally, also $SI_{4.1}$, which provides logic to show input error messages, is combined into the HTML file by means of Aggregation Operator 4. As with Aggregation Operator 3, the HTML code is adapted to load an additional JavaScript file, which contains the code of the input error message field as depicted with the top bold letters in the HTML snippet far right in Figure 7. Further, the visualization of the input error message field is put into the form so that it can show the validation results of the input fields. Of course, also this must be configured manually as only the web designer knows which error messages shall be displayed.

This use case shows that the concept of Solution Implementations can help to implement concrete solutions of several patterns together. Since user interfaces mostly incorporate many control elements, Solution Implementations can lead to immense reduction of effort in contrast to combine them manually. Especially if a developer has to deal with a vast of different technologies as mentioned above and, therefore, many specific implementation concepts for each of these technologies, Solution Implementations can provide a means to easily reuse available solutions for new use cases. Nevertheless, user interface design is often an act of creativity so that standardized implementations, as provided by Solution Implementations and Aggregation Operators, need to be adapted. But also in such cases, the presented concept can provide starting points with runnable code that then can be adapted creatively to meet the challenges of a non-standard user interface.

E. Use Case 5: Object-Oriented Software Engineering

General Use Case: A software engineer needs to combine an implementation of the Model View Controller Pattern with user interface patterns.

Concrete Scenario: An Aggregation Operator combines Solution Implementations of the Model View Controller pattern and the Pulldown Button Pattern in the form of Java classes. So, Solution Implementations from different pattern domains, i.e., different pattern languages are aggregated by means of an Aggregation Operators by adapting Java code.

When developing software systems, it is a common practice to first design the architecture of the software. In the architecture phase, design decisions are made, which are on an abstracter level in contrast to the concrete implementation problems, because they deal with general questions about the structure of software. In the domain of software architecture, patterns are a pervasive means to discuss design decisions and to describe the architecture of software systems [7]. They often affect later implementations, since the abstract structure of the software has to be implemented by concepts of the used technology. If Solution Implementations are provided for such patterns, the application of these patterns can be eased in order to save efforts to work them out manually for new use cases.

As already mentioned in the former use cases, patterns are also very common in the domain of user interaction design. Especially patterns describing control elements of user interfaces are often used. Thus, such patterns deal with problems that are very close to concrete implementations, since they often provide sketches that show how control elements should look like and how they should be arranged on a user interface [5].

This last use case shows how Solution Implementations of patterns from the two different domains of object-oriented design and user interaction design can be combined using our concept of Solution Implementations. Therefore, we show how an Aggregation Operator composes Solution Implementations of the pattern *Model View Controller (MVC)* [16], which is from the domain of object-oriented software architecture, and the *Pulldown Button* pattern [5], which is from the domain of user interaction design. The MVC pattern describes how the user interface of a program can be separated from its domain logic in order to prevent that changes of the user interface affect the implementation of the domain logic. Therefore, the user interface is encapsulated into a view entity, while the domain logic is provided by a model entity. The controller receives user interactions and triggers processing of domain logic based on the user's inputs. The pulldown button pattern provides a means to select exactly one value from a list of values. This list is only shown when a user clicks on the control element. If he or she selects a value from the list, the list is hidden again and only the selected value is visible.

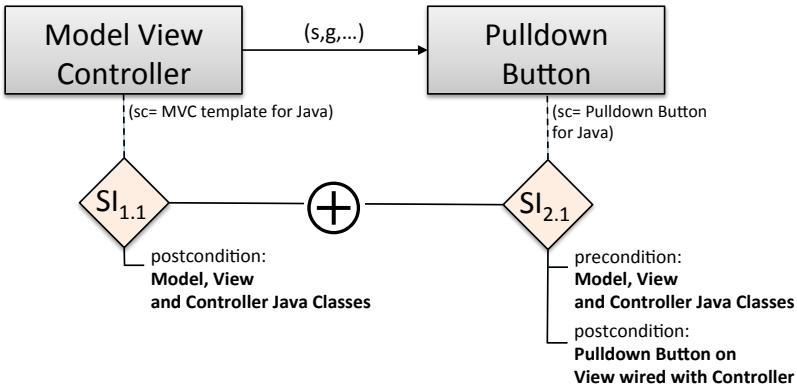
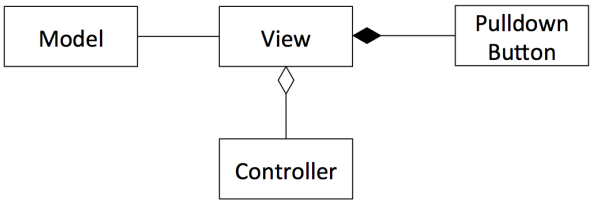


Figure 8. An Aggregation Operator combines Solution Implementations of the patterns Model View Controller and Pulldown Button.



Adaptions in View Class

```
public class myView extends JFrame{
     ...
     //define pulldown button
     JComboBox b;
     ...
     public myView(){
          ...
          //add pulldown button to view
          b = new JComboBox(valueList);
          this.add(b);
          ...
     }
     ...
}
```

Figure 9. Aggregated Solution Implementations of MVC and Pulldown Button in UML as well as adaptions of Java code by the Aggregation Operator.

Both patterns are depicted in Figure 8. For the sake of simplicity, there is just one Solution Implementation provided for each pattern – $SI_{1.1}$ and $SI_{2.1}$. Both Solution Implementations provide concrete solutions in the form of Java code as illustrated by the corresponding Selection Criteria. The postcondition of $SI_{1.1}$ "Model, View and Controler Java Classes" shows that this Solution Implementation consists of Java classes that implement the MVC paradigm. Further, the precondition of $SI_{2.1}$ matches the mentioned postconditions of $SI_{1.1}$, so both can be aggregated to form a combined solution.

The aggregation of both Solution Implementations is depicted as a Unified Modeling Language (UML) class diagram in Figure 9 [58]. The figure shows on the left that the pulldown button class is associated with the view class of the MVC Solution Implementation $SI_{1.1}$. To achieve this aggregation, the Aggregation Operator manipulates the java code of the view class so that an instance of the pulldown button is created and shown when the view is launched. Bold letters on the right in Figure 9 highlight the adaptions of the java code. So, this use case shows that the concept of Solution Implementations also allows combining solution knowledge from different pattern domains, since MVC is categorized as an architectural pattern, while pulldown button is a pattern from user interaction design. As they appear in different pattern languages, this use case demonstrates that Solution Implementations of patterns originally provided by different pattern languages can be applied together based on the presented approach. Of course, the aggregation must be adapted manually to place the pulldown button at the desired position and to select the appropriate view and so on. However, the actual aggregation, i.e., copying the corresponding java code, defining the required Java libraries, and linking the affected classes can be done by an Aggregation Operator automatically – and this already eases applying those patterns together in reality.

VI. PROTOTYPES

To prove the approach's technical feasibility, we implemented a pattern repository prototype that aims to capture patterns and their cross-references in a domain-independent way to support working with patterns [33][59]. Based on semantic wiki-technology, it enables capturing, management, and search of patterns. To adapt to different pattern domains, the pattern format is freely configurable. The pattern repository already contains various patterns from different domains such as cloud computing patterns [17], cloud data patterns [60], and costume patterns [12] to demonstrate the generic flexibility of our approach. The cross-references between the patterns enable an easy navigation through the pattern languages. Links like "apply after" or "combined with" connect the patterns, which results in a pattern language. The pattern repository does not only contain the patterns and their cross-references, but can be connected to a second repository containing Solution Implementations. We realized a Solution Implementation repository [33][61] for the domain of costume patterns to prove the interoperability of these two kinds of repositories. Here, for example, the concrete costumes of a sheriff occurring in a film are represented as the Solution Implementation of a sheriff costume pattern. By connecting the pattern to a Solution Implementation as a concrete solution of the abstracted solution of the pattern, the application of the pattern in a certain context is facilitated. Although the implemented solution repository for costumes in films is specifically tailored to store Solution Implementations from this domain, the concept of combining pattern repositories and solution repositories as described in [33] can easily be reused to create repositories for the other use cases to store code, HTML files, Cloud Formation Templates, or workflows.

To test the concept of Aggregation Operators, we prototyped the combination of several concrete Solution Implementations in the domain of cloud management patterns (use case 2). This domain is very appropriate, as the aggregation can be automated completely: we employed our workflow generator [41] to automatically combine different Management Planlets to an overall workflow implementing a solution to a problem that requires the use of multiple patterns. The input for this generator is a partial order of (composite) management planlets, i.e., Solution Implementations that have to be orchestrated into an executable workflow. This partial order is determined by the relations of combined patterns: if one pattern is applied after another pattern, also their Solution Implementations, i.e., Management Planlets, have to be executed in this order. The workflow generator creates BPEL-workflows while Management Planlets are also implemented using BPEL. As BPEL is a standardized workflow language, the resulting management plans are portable across different engines and

cloud environments supporting BPEL as workflow language, which is in line with the TOSCA standard [62][63][64]. Thus, this prototype shows that in certain domains, Aggregation Operators can be realized in an automated fashion. However, as seen in costumes, this is not always the case and in many other domains manual effort has to be spent for the aggregation.

VII. Conclusion and Future Work

In this paper, we introduced the concept of Solution Implementations as concrete instances of a pattern's solution. We showed how Solution Implementations can enrich patterns and pattern languages and how this approach can be integrated into a pattern repository. To derive concrete solutions for problems that require the application of several patterns we proposed a mechanism to compose these solutions from concrete solutions of the required patterns by means of Aggregation Operators. We concretized the general concept of Solution Implementations by five detailed use cases in the domains of cloud application architecture, cloud management, costumes in films, user interaction design and software engineering. We partially verified the approach by means of a prototype of an integrated pattern repository.

Currently, we extend the implemented repository for solution knowledge in the domain of costume design to capture Solution Implementations more efficiently. This repository integrates patterns and linked Solution Implementations in this domain and we enlarge the amount of costume Solution Implementations. We are also going to extend the presented approach to not only work on Solution Implementation sequences but also on aggregations of concrete solution instances not ordered temporally due to pattern sequences of a solution path. Since Solution Implementations are composed by Aggregation Operators, we are going to enhance our pattern repositories to also store and manage the Aggregation Operators. Finally, we will investigate Aggregation Operators in domains besides the above mentioned to formulate a general theory of Solution Implementations and Aggregation Operators.

ACKNOWLEDGMENT

This work was partially funded by the Co.M.B. project of the DFG under the promotional reference LE 2275/5-1.

References

[1] M. Falkenthal, J. Barzen, U. Breitenbücher, C. Fehling, and F. Leymann, "From pattern languages to solution implementations," Proceedings of the Sixth International Conference on Pervasive Patterns and Applications (PATTERNS), pp. 12–21, May 2014.

[2] C. Alexander, S. Ishikawa, M. Silverstein, M. Jacobson, I. Fiksdahl-King, and S. Angel, "A pattern language: towns, buildings, constructions," Oxford University Press, 1977.

[3] T. Iba and T. Miyake, "Learning patterns: a pattern language for creative learners II," Proceedings of the 1st Asian Conference on Pattern Languages of Programs (AsianPLoP 2010), pp. I-41 – I-58, March 2010.

[4] F. Salustri, "Using pattern languages in design engineering," Proceedings of the International Conference on Engineering Design, pp. 248–362, August 2005.

[5] M. van Welie, A pattern library for interaction design, http://www.welie.com, last accessed on 2014.11.28.

[6] R. Reiners, Bridge Pattern Library, http://bridge-pattern-library.fit.fraunhofer.de/pattern-library/, last accessed on 2014.11.28.

[7] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, "Pattern-oriented software architecture, volume 1: a system of patterns," Wiley, 1996.

[8] M. Fowler, "Patterns of enterprise application architecture," Addison-Wesley, 2003.

[9] T. Brunner and A. Zimmermann, "Pattern-oriented enterprise architecture management," Proceedings of the Fourth International Conference on Pervasive Patterns and Applications (PATTERNS), pp. 51–56, July 2012.

[10] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, "An architectural pattern language of cloud-based applications," Proceedings of the 18th Conference on Pattern Languages of Programs (PLoP), pp. A-20–A-30, October 2011.

[11] J. Yoder and J. Barcalow, "Architectural Patterns for Enabling Application Security," Pattern Languages of Program Design 4, pp. 301–336, 2000.

[12] D. Schumm, J. Barzen, F. Leymann, and L. Ellrich, "A pattern language for costumes in films," Proceedings of the 17th European Conference on Pattern Languages of Programs (EuroPLoP), pp. C4-1–C4-30, July 2012.

[13] PHP, PHP: Hypertext Preprocessor, http://php.net, last accessed on 2014.11.28.

[14] E. Gamma, R. Helm, R. Johnson, and J. Vlissides, "Design patterns: elements of reusable object-oriented software," Addison-Wesley, 1995.

[15] G. Hohpe and B. Wolf, "Enterprise integration patterns: designing, building, and deploying," Addison-Wesley, 2004.

[16] T. Reenskaug, "The original MVC reports," https://heim.ifi.uio.no/~trygver/2007/MVC_Originals.pdf, last accessed on 2014.11.28.

[17] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, "Cloud computing patterns," Springer, 2014.

[18] F. Buschmann, R. Meunier, H. Rohnert, P. Sommerlad, and M. Stal, "Pattern-oriented software architecture volume 1: a system of patterns," Wiley, 1996.

[19] R. Reiners, R. Halvorsrud, A. Wegner Eide, and D. Pohl, "An approach to evolutionary design pattern engineering," Proceedings of the 19th international Conference on Pattern Languages of Programs, October 2012, scheduled for 2014.

[20] U. Zdun, "Systematic pattern selection using pattern language grammars and design space analysis," Software: Practice and Experience, vol. 37, pp. 983–1016, 2007.

[21] R. Reiners, "A pattern evolution process – from ideas to patterns," Lecture Notes in Informatics – Informatiktage 2012, pp. 115–118, March 2012.

[22] M. Falkenthal, D. Jugel, A. Zimmermann, R. Reiners, W. Reimann, and M. Pretz, "Maturity assessments of service-oriented enterprise architectures with iterative pattern refinement," Lecture Notes in Informatics - Informatik 2012, pp. 1095–1101, September 2012.

[23] R. Porter, J. O. Coplien, and T. Winn, "Sequences as a basis for pattern language composition," in Science of Computer Programming, Special issue on new software composition concepts, vol. 56, pp. 231–249, April 2005.

[24] C. Fehling, F. Leymann, R. Mietzner, and W. Schupeck, "A collection of patterns for cloud types, cloud service models, and cloud-based application architectures," http://www.cloudcomputingpatterns.org, last accessed on 2014.11.28, University of Stuttgart, Report 2011/05, Mai 2011.

[25] U. van Heesch, Open Pattern Repository, https://code.google.com/p/openpatternrepository/, last accessed on 2014.11.28.

[26] M. Demirköprü, "A new cloud data pattern language to support the migration of the data layer to the cloud," in German "Eine neue Cloud-Data-Pattern-Sprache zur Unterstützung der Migration der Datenschicht in die Cloud," University of Stuttgart, diploma thesis no. 3474, 2013.

[27] C. Fehling, F. Leymann, J. Rütschlin, and D. Schumm, "Pattern-based development and management of cloud applications," Future Internet, vol. 4, pp. 110–141, 2012.

[28] C. Fehling, F. Leymann, R. Retter, D. Schumm, and W. Schupeck, "An architectural pattern language of cloud-based applications," Proceesings of the 18th Conference on Pattern Languages of Programs (PLoP), pp. A-20 – A-21, Oct. 2011.

[29] A. G. Mirnig and M. Tscheligi, "Building a general pattern framework via set theory: towards a universal pattern approach," Proceedings of the Sixth International Conference on Pervasive Patterns and Applications (PATTERNS), pp. 8–11, May 2014.

[30] D. Krleža and K. Fertalj, "A method for situational and guided information system design," Proceedings of the Sixth International Conference on Pervasive Patterns and Applications (PATTERNS), pp. 70–78, May 2014.

[31] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Automating cloud application management using management idioms," Proceedings of the Sixth International Conference on Pervasive Patterns and Applications (PATTERNS), pp. 60–69, May 2014.

[32] J. Barzen and F. Leymann, "Costume languages as pattern languages," accepted at Pursuit of Pattern Languages for Societal Change, unpublished.

[33] C. Fehling, J. Barzen, M. Falkenthal, and F. Leymann, "PatternPedia - collaborative pattern identification and authoring," accepted at Pursuit of Pattern Languages for Societal Change, unpublished.

[34] C. Alexander, "The timeless way of building," Oxford University Press, 1979.

[35] Amazon, AWS Cloud Formation, http://aws.amazon.com/cloudformation/, last accessed on 2014.11.28.

[36] Amazon, Elastic Beanstalk, http://aws.amazon.com/elasticbeanstalk/, last accessed on 2014.11.28.

[37] Amazon, Amazon Web Services, http://aws.amazon.com, last accessed on 2014.11.28.

[38] Microsoft, Microsoft Azure, http://azure.microsoft.com, last accessed on 2014.11.28.

[39] Amazon, AWS EC2, http://aws.amazon.com/ec2/, last accessed on 2014.11.28.

[40] C. Fehling, F. Leymann, S. T. Ruehl, M. Rudek, and S. Verclas "Service migration patterns – decision support and best practices for the migration of existing service-based applications to cloud environments," Proceedings of the IEEE International Conference on Service Oriented Computing and Applications (SOCA), in press, December 2013.

[41] U. Breitenbücher, T. Binz, O. Kopp, and F. Leymann, "Pattern-based runtime management of composite cloud applications," Proceedings of the 3rd International Conference on Cloud Computing and Service Science (CLOSER), pp. 475–482, May 2013.

[42] U. Breitenbücher, T. Binz, O. Kopp, F. Leymann, and M. Wieland, "Policy-Aware Provisioning of Cloud Applications," in SECURWARE. Xpert Publishing Services, August 2013, pp. 86–95.

[43] O. Kopp, H. Eberle, and F. Leymann, "The subprocess spectrum," Proceedings of the 3rd Business Process and Services Computing Conference (BPSC), pp. 267–279, September 2010.

[44] J. Tidwell, "Designing interfaces – patterns for effective interaction design," O'Reilly, 2011.

[45] D. K. van Duyne, J. A. Landay, and J. Hong, "The design of sites: patterns for creating winning websites," Prentice Hall, 2007.

[46] J. Borchers, "A pattern approach to interaction design," John Wiley & Sons, 2001.

[47] Yahoo Developer Network, Yahoo design pattern library, https://developer.yahoo.com/ypatterns/, last accessed on 2014.11.28.

[48] World Wide Web Consortium, HTML 4.01 Specification, http://www.w3.org/TR/html401/, last accessed on 2014.11.28.

[49] Ecma International, ECMAScript Language Specification, http://www.ecma-international.org/ecma-262/5.1/, last accessed on 2014.11.28.

[50] Oracle, Java Servlet Technology, http://www.oracle.com/technetwork/java/index-jsp-135475.html, last accessed on 2014.11.28.

[51] Oracle, JavaServer Pages Technology, http://www.oracle.com/technetwork/java/javaee/jsp/index.html, last accessed on 2014.11.28.

[52] Oracle, JavaServer Faces Technology, http://www.oracle.com/technetwork/java/javaee/javaserverfaces-139869.html, last accessed on 2014.11.28.

[53] Google, Angular.js, https://angularjs.org, last accessed on 2014.11.28.

[54] jQuery, jQuery, http://jquery.com, last accessed on 2014.11.28.

[55] Spring, Spring Framework, http://projects.spring.io/spring-framework/, last accessed on 2014.11.28.

[56] Rails, Ruby on Rails, http://rubyonrails.org, last accessed on 2014.11.28.

[57] Google, Google Web Toolkit, http://www.gwtproject.org, last accessed on 2014.11.28.

[58] Object Management Group, Unified Modeling Language, http://www.uml.org, last accessed on 2014.11.28.

[59] N. Fürst, "Semantic wiki for capturing design patterns," in German "Semantisches Wiki zur Erfassung von Design-Patterns," University of Stuttgart, diploma thesis no. 3527, 2013.

[60] S. Strauch, V. Andrikopoulos, U. Breitenbücher, S. Gómez Sáez, O. Kopp, and F. Leymann, "Using patterns to move the application data layer to the cloud," Proceedings of the 5th International Conference on Pervasive Patterns and Applications (PATTERNS), pp. 26–33, May 2013.

[61] D. Kaupp, "Application of semantic wikis for solution documentation and pattern identification," in German "Verwendung von semantischen Wikis zur Lösungsdokumentation und Musteridentifikation," University of Stuttgart, diploma thesis no. 3406, 2013.

[62] OASIS, Topology and Orchestration Specification for Cloud Applications Version 1.0, http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.html, last accessed on 2014.11.28.

[63] T. Binz, U. Breitenbücher, O. Kopp, and F. Leymann, "TOSCA: portable automated deployment and management of cloud applications," in Advanced Webservices, A. Bouguettaya, Q. Z. Sheng, F. Daniel, Eds., Springer, pp. 527–549, 2014.

[64] U. Breitenbücher, T. Binz, K. Képes, O. Kopp, F. Leymann, and J. Wettinger, "Combining Declarative and Imperative Cloud Application Provisioning based on TOSCA," in IC2E. IEEE, March 2014, pp. 87–96.

Introducing a Scalable Encryption Layer to Address Privacy and Security Issues in Hybrid Cloud Environments

Paul Reinhold and Wolfgang Benn
Chemnitz University of Technology
Chemnitz, Germany
Email: {paul.reinhold@s2012, wolfgang.benn@informatik}.tu-chemnitz.de

Benjamin Krause and Frank Goetz
Qualitype GmbH
Quality Management Systems
Dresden, Germany
Email: {b.krause,f.goetz}@qualitype.de

Dirk Labudde
Hochschule Mittweida
University of Applied Sciences
Mittweida, Germany
Email: dirk.labudde@hs-mittweida.de

***Abstract*—Besides security and privacy concerns, high efforts, necessary for developing and maintaining cloud services in the modern IT landscapes, are new major issues in small and medium enterprises. Software service development and operations face new challenges in dynamic cloud environments. To establish the cloud service and to address privacy and security issues, our suggested scalable Encryption Layer can be used. With our approach, small and medium enterprises, can securely outsource their data to public cloud storages preventing the public cloud provider from data insight, even with full access to the physical machines. This paper introduces an in-depth description for setting up our Encryption Layer, and also provides solutions for problems which may arise during the implementation and setup process of cloud environments. The presented test results of our implemented prototype demonstrate, with an overall overhead of 50% to 75%, the practical applicability.**

Keywords-Hybrid Cloud; Cloud Security; Architecture Layer; Industry Research; Small Medium Enterprises.

I. INTRODUCTION

The increasing knowledge about cloud computing technology and its publicity leads to a growing number of service offerings over the Internet. Even small and medium sized enterprises (SME) are able to offer services for a large number of consumers through cloud computing concepts. Like previous work [1] shows, with the usage of an appropriate hybrid cloud concept, SME can develop practical cloud service solutions with respect to privacy and security.

The high acceptance of services, like Instagram [2] or Dropbox [3] for private usage, suggests that private consumers have a lower privacy demand than business users. Studies of Gens, like [4], [5], support this hypotheses. A recent study of BITCOM [6] indicated that in Europe, and especially in Germany, the acceptance of public cloud services for business purposes is low. Typical reasons are security and privacy concerns. The most recent study of Crisp Research [7] has led to the same results. This study is most interesting for our work, since it mainly focused on SMEs. The study showed that around 40% of the respondents hesitate with cloud solutions, because their customers have concerns about security and privacy. Interestingly, with a percentage of around 60%, the main arguments against cloud computing solutions are the following: First, the high effort to run a cloud service and second, the high costs for developing a new cloud application. This shows, despite the security and privacy issues, SME hesitate because of deficient resources, like hardware, men power or know-how. Therefore, we aim to lower these concerns by introducing a convenient additional architecture layer, called *Encryption Layer* (EL). In a previous work [1] we demonstrate the practicability and low migration effort of this EL into existing software systems. In this paper we focus on a detailed insight of the implementation process and the effort to set up the appropriate environments. Our suggested EL is located between business logic and persistence layer. For evaluation purposes, we implement a prototype using the suggested architecture to outsource unstructured (files) and structured (databases) data into a public cloud in an encrypted and secured manner. For that purpose, our work focuses on SME providers that already run services or web applications and take new cloud offerings into account in an effort to become more cost-efficient, or to establish new business models, as discussed in [7]. In addition, SME providers probably use own hardware to run their services and plan to develop a new version or new service, which would exceed the current limit of their hardware. Another scenario is that the provider needs to invest in new hardware to keep its services running and is looking for lower cost alternatives.

The outline for the rest of the paper is as follows: In Section II, a comparison of common cloud delivery models with respect to privacy, costs and performance is discussed. Based on this comparison we elucidate the possible solutions with a reasonable effort for SME. In Section III, an abstract overview of our solution, as well as some differences to other hybrid cloud approaches is given. In addition, we discuss the applicability of a key management system inside of hybrid cloud environments. As the main part of the paper, Section IV summarizes the EL prototype implementation, and gives in-depth technical insights, as well as an evaluation of tests scenarios and results. Especially, we describe some pitfalls and typical problems, which may arise during the development process and appropriate solutions to avoid them. In Section V, a critical discussion of the test results is provided, as well as pros and cons of an additional layer, like EL, in general. In Section VII, we elucidate related and future works. Finally, Section VIII concludes the paper.

TABLE I. COMPARISON OF DIFFERENT CLOUD MODELS FROM CONSUMERS AND PROVIDERS POINT OF VIEW

View	Criteria	Private Cloud	Public Cloud	Hybrid Cloud
consumer	cost	high	low	medium
	privacy	medium	low	high
	data-at-rest encryption	yes	yes	yes
	key owner	provider	provider	consumer (and provider)
	key management	by provider	by provider	by provider (and consumer)
provider	cost	very high	low	medium
	availability	medium	very high	high
	backup	medium	very high	very high
	hardware needs	high	very low	medium
	effort to run service	very high	low	medium - high
	flexibility	high, but limited	very high	higher, but limited
	scaling	yes, but limited	yes	yes, but limited

II. CLOUD DEPLOYMENT MODEL COMPARISON

In the following sections, the end-user of a cloud service shall be named cloud consumer or simply consumer [8]. From the consumers view, the provider offers services over the Internet. Whether the service is offered by provider's hardware or by third party resources is irrelevant for the customer, as long as service supply is ensured. However, the method of providing can be essential for the acceptance of the service on consumers side.

According to literature [9], [10], [11], [12] most common cloud deployment models are private, public and hybrid cloud models. In the private cloud, the provider runs its own cloud. As Rhoton and Haukioja [13] mentioned some would argue that anything less than a full cloud model is not cloud computing. Actually, private cloud computing contradicts the idea of cloud computing through limitations in basic cloud characteristics defined by the National Institute of Standards and Technology (NIST) in [9] like rapid elasticity, on-demand self-service and resource pooling. Nevertheless, the term is widely accepted in academia and industry. Private cloud providers own the hardware and have exclusive access to it. To leverage cloud effects the provider runs its hardware in form of a cloud, allowing flexibility and scaling. Public clouds are offered by public cloud service providers (CSP). In contrast to private clouds, this form uses all advantages of cloud computing. Therefore, public clouds are the most flexible and cost-efficient services, since resources are obtained by need and payed by usage. As Fernandes et al. [14] mentioned, this model is less secure and more risky than other deployment models. Actually, this statement is trivial to confirm. Since public cloud computing is an extensive form of IT outsourcing, there is a much higher potential for malicious attacks compared to private cloud solutions. Including not only external attacks but internal attacks, e.g., through malicious administrators as well.

Hybrid clouds are the third common approach where services run both in a private and public clouds. We think this is the most interesting approach, having a high potential for balancing cloud computing advantages versus security and privacy issues. In addition, this approach seem the most likely one for using cloud computing in SME, van Hoeck et al. [15] supports this hypothesis. In our opinion the main aspect is to use private (expensive) resources as little as needed and public (cheap) resources as much as possible. However, this optimization is a highly complex task, especially for existing enterprise IT infrastructures or software systems.

Table I shows a comparison between the three cloud deployment models from both consumer's and provider's point of view. An actual survey on cloud security carried out by Fernandes et al. [14] has shown similar results. The costs factor for both consumer and provider is comprehensible, since hardware expenses are usually passed to consumers.

Privacy is low for public and medium for private cloud architecture. Authors, like Wang and Jia [16], argued that the private cloud could provide the the highest degree of security for users data. We do not fully agree with that, because this depends on who is the owner of the cloud and whose data is processed or stored in this private cloud. If data and cloud owner are the same, Wang and Jias [16] argument might be right. However, in our scenario, as well as in most cloud service offerings, the private cloud owner differs form the data owner. Private clouds often have strong authorization and access control concepts, but no special requirement to secure data with encryption against the provider (SME) itself [17]. Thus, the private cloud provider often has access to customer data and their customers data, respectively. Therefore, the argument of Wang and Jia [16] is not true in our scenario. In a public cloud it is very costly or impractical to secure data and to keep them available for processing at the same time, which, for instance, fully homomorphic encryption [18] can provide. However, approaches like Mylar [19] can pose an alternative, by the use of encryption in client software. Nevertheless, this cause additional tasks, like key management, suggesting a hybrid cloud approach offers a more attractive solution with respect to the customer demands. Another important aspect for consumers is data-at-rest encryption. This form of encryption is possible in all of the models, but implies tasks for key ownership and key management. If the same instance encrypts data and stores the referring key, no trustable security can be guaranteed, because providers can decrypt data without consumers knowledge or permission. This has been recently documented for economically rational cloud providers [20]. Because of this circumstance hybrid clouds suggest a solution where consumers get more control over their data and the possibility for public cloud providers to access unencrypted files is eliminated. Even if a consumer trusts its provider (with a private cloud) and consequently encryption is not needed, the hybrid solution is more economical. From the providers point of view, a private cloud can not provide the availability as it is guaranteed by a public cloud. The hybrid model benefits from this fact by outsourcing parts of the software

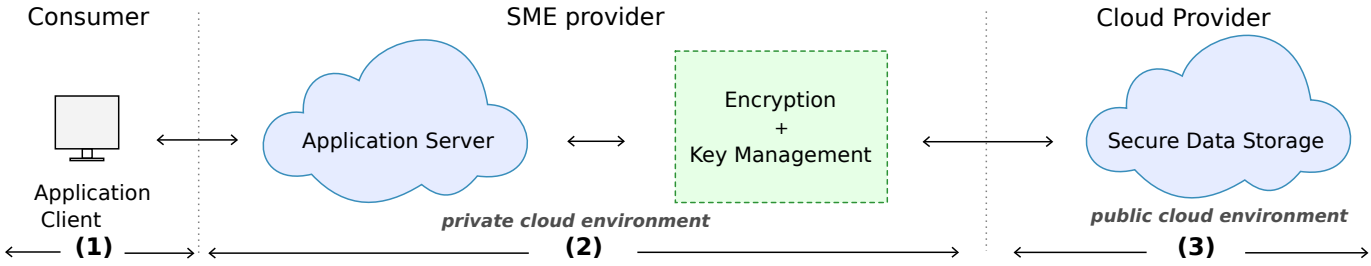


Figure 1. The hybrid cloud environment with Application Client, Application Server and outsourced Data Storage. The green dashed box is our additional Encryption Layer, located between the application server in the private cloud and data storage in the public cloud.

solution in a public cloud. Backup security underlays the same principle, in fact the backup process in the hybrid model can be outsourced completely. The hardware needs and the effort to run the service are coherent. Lots of own hardware means not only to manage, but also to maintain and have environment settings (buildings, redundant broadband internet access) to run a private cloud. As mentioned above, the great advantages of cloud computing like flexibility and scaling are limited in private and hybrid cloud solutions. As a result of this comparison and questions, our aim in [1] was to combine the security of a private cloud with the flexibility, reliability and availability of a public cloud, creating a balanced solution.

III. Hybrid Cloud Environments

The constellation of a hybrid cloud computing environment is illustrated in Figure 1. It consists typically of a client (1), a private (2) and a public cloud component (3). The most common usage of a hybrid cloud environment in literature [21], [22], [16], [23], [24] is a separation of sensitive and non-sensitive data, stored in private and public cloud, respectively. Lot of research has been done for developing efficient classification algorithms to separate or sanitize sensitive data from non-sensitive data. The aim is to outsource as much data as possible in public cloud. Our approach differs from this kind of hybrid cloud computing. By the use of encryption methods we outsource all kind of data, regardless their sensitivity. Resulting advantages are: no need for data labeling/classification, less effort to integrate in existing systems, and a less complex and therefore less error-prone overall system architecture.

A. Private cloud environment

As shown in Figure 1 the private cloud environment consists of application server and is managed by the SME provider. That means all business logic remains inside the private cloud, resulting in the main disadvantage of our hybrid cloud solution. Actually, there is no alternative if neither consumer nor SME provider trust a public cloud provider anyway. Since efficient data-in-use encryption is still an open issue; even with great improvements like fully homomorphic encryption [18]. Consequently, our approach uses the existing system of the SME provider, adding an additional architecture layer (green dashed box) for protecting data in the public cloud with encryption methods.

B. Public cloud environemt

The public cloud component is used for data storage in form of virtual servers provided by an public IaaS provider. As Rhoton [13] mentioned, this is the most basic form of using cloud computing resources. We do not consider storage solutions like S3 [25] or Azure SQL Database [26] to avoid vendor lock-in effects and be more flexible. However, as we address SME with this work the establishment of a basic, yet efficient cloud solution (as a first step) is our focus. As the cloud provider selection shows, there is no reason to use one public cloud provider exclusively. This is an extended form of hybrid cloud computing, called multi-cloud or multi-source solutions, also described by Bohli et al. [27] and Li et al. [28]. Actually, the classification if its a hybrid or a public cloud solution depends mainly on the point of view. As mentioned, most cloud solutions in practice are hybrid (multi-) cloud ones.

C. Hybrid key management system

Similar to the identity management system classification described by Hussain [29], we can separate key management systems in user centric and federated systems. While a user centric key management results in a high overhead for the consumer, by keeping the keys in a secure way, this is outsourced in federated case to a third party or the provider of the service. Just like the hybrid deployment model, we can use a hybrid key management model. The user keeps a master key as the root of an encrypted key tree, like described by Zarandioon et al. [30], while the tree is managed by the SME. As a result the SME can not read out plaintext data without the consumer's permission. Figure 2 illustrates the basic idea. In fact, a similar concept is used by Apple's instant messenger service iMessage [31]. Despite some privacy issues described by QuarkLabs [32], in their opinion this cloud based instant messenger service could be considered as the most practical and secure real-time messaging system available. As a result, we think a hybrid key management system, combined with state-of-the-art cryptography can be considered as highly realistic for practical use in future work for our approach.

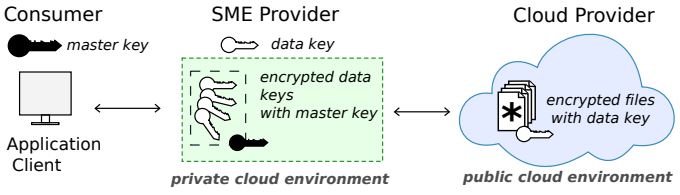


Figure 2. The hybrid key management system consists of a master key stored by the consumer and data keys stored by the SME provider. The data keys are encrypted by the consumer's master key.

IV. Scalable Encryption Layer

The scalable encryption layer is an additional tier between the application servers and the data storage server in the public cloud. Its location is illustrated by the green dashed box in Figure 1. As the figure shows the EL consists of two components, the encryption and the key management service. These services extend the proxy-like behavior of the CryptDB MySQL proxy developed by Popa et al. [33]. The basic idea of our EL is the encryption and decryption of data while transferring the data to the outsourced data storage (on-the-fly encryption). Therefore, less trust to the party to which the data has been outsourced is needed. To prevent the EL becomes kind of a bottle neck, it is located in a private cloud. This private cloud is controlled and managed by the SME provider and offers an efficient possibility for scaling.

Nevertheless, this approach is accompanied with some issues. First of all, virtual cloud instances should be as stateless and independent as possible. So, how and where to persist the keys need for encryption and decryption? Second, because of scaling and load balancing the following scenario is very common: Instance A encrypts some file, but the decryption request is forwarded to instance B. How to exchange encryption keys, if the instances should be independent? Third, the encryption layer should support the encryption of both structured and unstructured data. Which instance should decide which type of data is send to the EL and how it should be forwarded (in sense of load balancing). Fourth, despite the highly dynamic cloud environment there has to be some kind of a cloud access point, the application server can address their requests to. In this scenario this access point needs to know what instances exist, to be able to forward requests to them. Last but not least, all these problems and requirements should be solved in a way so that the existing system has to be adjusted as little as possible (low migration effort). Summing up, there are the following problems to solve:

1) Persistence and distribution of encryption keys.
2) Load Balancing, request forwarding with respect to the type of data.
3) Cloud instance controlling and addressing.
4) Low migration effort.

A. Technical setup and implementation of the protoype

The very first point to clarify is, which cloud management system is used for the private cloud. Important to mention is, this cloud management system only provides the cloud environment. That means, especially in the field of scaling, there are tools to easily start and stop virtual instances. However, in which way the implemented system reacts to upscaling (include for load balancing) or downscaling (exclude from load balancing) events, has to be managed by the system itself. Giving some thoughts, this fact is very clear, in an IaaS cloud environment the management system usually does not know what is inside these virtual instances and can therefore not react in any specific behavior. This is the main difference to PaaS solutions, where the management gets much more context information from source code, deployment rules and configuration files. In consideration of these facts a PaaS solution would be the favorable solution, but the effort to setup this solution is much too high for this prototype. So, we decided to use OpenNebula 4.4 [34] for our private cloud environment. Reasons for this decision are the open source environment, good possibility for own integrations and, last but not least, the fact of existing knowledge about OpenNebula. Our private cloud is powered by four physical hosts with 3 GHz Dual-Cores and 8 GB RAM. These machines consist of standard components and are connected via a common 100 MBit/s ethernet network to keep the hardware costs low.

After installation and setup the environment we had to cope with a very basic problem, to which we refer to as *image persistence dilemma*. Basically, there are two possibilities to setup an image in OpenNebula, from which the virtual instances are created: First kind are persistent images. As the name implies, these images save the adjustments the user is doing during runtime of the virtual machine. Besides this, another advantage of this *stateful* image is the rapid boot time. However, there are some crucial disadvantages. First of all, the access is exclusive, which means there is no possibility for scaling, based on persistent images. This makes perfect sense in consideration of constancy. Secondly, because of the direct usage of the image in the cloud-internal image repository (e.g., SAN/NAS as possibilities to provide access to a data storage in networks), which results in fast boot times, but the runtime performance is lower than in non-persistent images. This is because of the higher access time for the network storage, in contrast to the local disk, resulting in a higher CPU *wait time*. Especially, we observe this behavior by execution of write heavy disc access tasks, e.g., compiling the CryptDB MySQL proxy.

The second kind are non-persistent images. Again, as the name implies, all changes done while runtime inside this VM, are lost when shutting down this VM. In addition, the boot time for this kind of images are much longer, because the host needs a local copy on its physical hard drive. Trivially, a significant amount of time is necessary to transfer an image of 10-20 GB over the network. The advantages of non-persistent images are the better runtime performance and, even more important, the possibility to have more than one virtual instance of this image. One can argue, that long boot time only happens once, because the second instance of the host could copy the local image. However, there are two important points. First of all the possibility of a local copy depends on the image format. Some formats, like *qcow* [35], support copy on write. That means the local changes were stored in a different place, which opens up the possibility for other virtual instances to use the same local image. However, the hypervisor installed in the physical host, must be aware of this. Although we used the qcow format, the images were copied again from the image repository. At this point we see potential for future work. The second point is in consideration of the performance and reliability it would be better to run the other instances of the image on as many different (physical) hosts as possible.

How to solve this image persistence dilemma? Actually, the public IaaS cloud provider we use for data storage server provides a simple solution. There are only persistent images. So, if you want to scale up, you have to clone your images as often you want to scale up. This is inefficient in a lot of ways. First, it takes a lot of storage, resulting in high costs and inefficiency, because most of the time the images are not used. Second, there is an up-scaling limit in short-term by the number of cloned image. Third, because of the different images there are a lot of different states within the same virtual instances, making potential failure analysis extremely difficult.

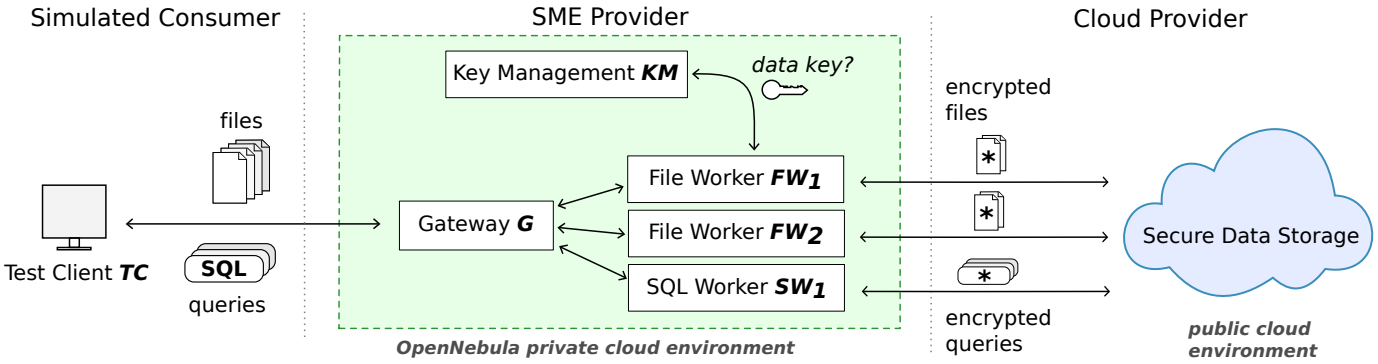


Figure 3. The implemented encryption server prototype with the Test Client TC to simulate consumer requests (files and SQL queries), our Encryption Layer (green dashed box) to encrypt/decrypt consumer requests and the data storage to persist the encrypted files/databases. The data flow is illustrated as well, showing only encrypted data leaves the private cloud environment.

However, in this prototype we have no need for scaling in the public cloud, but for productive systems this should be a crucial point while choosing a public cloud provider.
All in all, there is no perfect solution of the image persistence dilemma. The most common solution is the usage of a cluster file system, distributed over the hosts with high speed and high bandwidth internal network. This is open for future work. It has to be mentioned, that this optimizations toward a highly performant cloud environment is not suitable for most SME. As we can see, even public cloud providers lack in providing efficient and practical solutions. Therefore, the practical suitability of setting up a highly efficient and performant private cloud environment is not included in the practicability discussion of the EL.

1) Setup of the cloud system - virtual machines: Figure 3 shows an overview of the architecture of the implemented prototype, with the Test Client TC, the EL and the cloud storage. As mentioned, the EL consists of a key management and an encryption component. The focus of our work is setting up and implementing the encryption part as the core component and as crucial point for practicability discussions for SME SaaS provider. Therefore, the key management server KM is a basic MySQL database server for storing the encryption keys for file encryption and a basic NFS server for the SQL query encryption. Despite the fact this is not suitable for productive systems, it fulfills the persistence problem in a very efficient way and is therefore a good solution for our prototype implementation. Because the keys must be stored persistently and KM will not scale, it runs as a VM based on a persistent image.
As Figure 3 shows, besides KM the EL consists of a Gateway G, File Workers FW_i and a SQL Worker SW_1. With the decision of gateway solution, we address the problems of load balancing, forwarding, cloud instance controlling and addressing, and the low migration effort, by the risk of a single point of failure. In addition, the workers become more stateless and independent from application servers. We think for the prototype implementation this tradeoff is suitable. The gateway acts as a load balancer based on a JBoss AS 7 cluster, which also solves possible problems with internal communication between highly dynamic VMs. It also acts as a MySQL proxy, forwarding all database requests to the SQL Worker SW_1. As a result, the gateway behave for the application server (Figure 1) as a file server and database. This fact reduces migration effort significantly. Because the gateway does not need to store any data permanently, the image is non-persistent. Actually, there is the possibility to scale up the gateway. However, our hardware setup was not suitable to test such large scenario.
Another non-persistent image is used by the File Worker FW_1. These workers perform the encryption and decryption task and act as stateless and scalable nodes in a JBoss Cluster. As Figure 3 shows these workers request encryption keys from the key management server and upload files into the public cloud.
We ran in the problem how to deploy an application archive in the JBoss nodes. Because it is very inefficient to shut down FW_1, set the image to persistent state, start it again, deploy the archive, shut down, and switch back to non-persistent state to make it scalable again. After that we can start up the service again. Our solution will instead of pushing the new archive version into the virtual machine, pull it while starting up the VM. This is possible by the script shown in listing 1, which is processed while booting up the VM.

Listing 1. Application archive update/retrieval script

```
curl -Lkv -o /tmp/servlet.war -u <name>:<pw> '<URL>'
rm <JBOSS_DEPLOY_DIR>/servelet*
mv /tmp/servlet.war <JBOSS_DEPLOY_DIR>servlet.war
```

The first command fetches the servlet archive from the given $< URL >$. In our case, we get the latest version of the application archive stored on our internal repository management system for software artifacts (Sonartype Nexus [36]). Second and third command remove a possible old version and move the new application archive into the deployment directory of the JBoss application server. Actually, this naturally enables us to update an application while runtime, just by restarting a VM (or by triggering the script manually at VM runtime, therefore, the JBoss hot-deploy mechanism has to be active).
The SQL Worker SW_1 as the fourth image is also non-persistent. Initially, we plan to scale the SQL worker as well. However, it was not possible to outsource the key management of the CryptDB-enabled MySQL proxy deployed in SW_1 to KM with reasonable effort. Therefore, we outsource the database files of the internal database of the CrytptDB via NFS to KM. Trivially, if more than one SQL worker would (over)write these database files the consistency could not be ensured. As result we could not scale the SQL Worker.
All VMs are part of an autoscaling service of the OpenNebula cloud environment called *OneFlow*[37]. This service allows to

bundle some VMs and to set up a set of rules for scaling the virtual machines. OneFlow enables to adjust the number of VMs at the startup and how much load a VM has to have to scale up or down. Starting up our service with these five VMs takes around 10 min.

2) Setup of the JBoss Cluster: Figure 4 shows the structure of the internal JBoss Cluster, as described by Marchioni [38]. This cluster is used for load balancing in a dynamic environment. The cluster consists of a controller, which is basically an extended HTTP server located in the Gateway VM G. Another part of the cluster are nodes, that are configured JBoss AS 7 application server. The JBoss cluster addresses problems 2) Load Balancing and 3) Cloud instance control mentioned at the beginning of Section IV. So, this cluster is perfectly suited for the management of dynamically added and removed virtual machines. However, there is a limitation. As the HTTP server based controller suggests, the cluster only supports HTTP request. Therefore, it can not be used to load balance SQL query requests for the SQL Worker SW_1 and works only for the File Workers FW_i. Though, in consideration of the fact that only FW_i are scalable, this is perfectly fine.

The JBoss cluster is based on mod_cluster 1.2.6 [39]. The setup of the JBoss cluster consists of two main aspects: setting up the controller and setting up the nodes. As mentioned, the controller is located in G. To make the standard apache 2 HTTP server work as a JBoss cluster controller, an extension by the mod_cluster module is necessary. For configuration details please see [40]. To setup a standard JBoss AS7 as a cluster node, it is necessary to enable the mod_cluster module and configure it appropriately. Especially the multicast ports have to be the same as configured in the controller in G. The cluster communication is illustrated in Figure 4 and works like following. The controller in G sends out a multicast, containing the controller address information. As the nodes receive this multicasts they answer to the appropriate controller address, containing node information like deployed application archives and node load-metrics. Finally, the controller receives these node answers and can take them into account for load balancing. This principle is perfectly suited for dynamic cloud environments, as nodes can start and stop at any time.

For load balancing, mod_cluster provides a lot of load metrics. Our metrics are shown in the listing 2. For detailed meaning of the parameters please see [41].

Listing 2. Load balancing configuration

```
<dynamic-load-provider history="10" decay="2">
 <load-metric type="cpu" weight="2" capacity="1"/>
 <load-metric type="mem" weight="4" capacity="512"/>
 <load-metric type="ST" weight="1" capacity="512"/>
 <load-metric type="RT" weight="1" capacity="512"/>
</dynamic-load-provider>
```

G uses this metrics to calculate the busyness b of the node (weighted average), after the equation,

$$b = \frac{2 * cpu + 4 * mem + ST * RT}{8} \quad (1)$$

in which ST stands for *send-traffic* and RT stands for *received-traffic*. It has to be pointed out that the cpu metric in our virtual environment does not behave in the intended manner. We cloud not evaluate the exact reason, but numerous tests show that the configuration above leads to better results than one without the cpu metric.

In order to establish a high data security, we chose an at least 256 bit standard encryption method. Therefore, we have to use another Java security provider, because the standard security provider of Java only supports up to 128 bit encryption key length. So, we decided to use the security provider of Bouncycastle [42]. To use it in the server components in the nodes, we have to extend the JBoss AS7 nodes. First of all we replace the standard Java policy files with those from Bouncycastle. By default it is located under a path like *JAVA_HOME_DIR\jdk1.7.x_xx\jre\lib\security*. Secondly, a new module has to be added in the JBoss AS7 nodes. Therefore, the creation of a folder like *JBOSS_HOME_DIR/modules/org/bouncycastle/main* is necessary. The Bouncycastle library files have to be moved in this folder. The next step is the creation of the *module.xml* as shown in listing 3.

Listing 3. Bouncycastle module

```
<module xmlns="urn:jboss:module:1.1"
name="org.bouncycastle">
 <resources>
 <resource-root path="bcprov-jdk15on-150.jar"/>
 </resources>
 <dependencies>
 <module name="javax.api" slot="main"export="true"/>
 </dependencies>
</module>
```

The last step is to integrate this module, as listing 4 shows, in the JBoss AS7 server configuration file. One example could be *JBOSS_HOME_DIR/standalone/configuration/standalone.xml*.

Listing 4. JBoss AS7 server configuration

```
<global-modules>
      <module name="org.bouncycastle" slot="main"/>
</global-modules>
```

Summing up, to set up the JBoss Cluster as appropriate environment for the EL consists of three steps. The basic setup with controller in the gateway G and nodes in FW_i. Second, the configuration of the load balancing metrics and third, the extensions of the nodes to support 265 bit encryption standard methods.

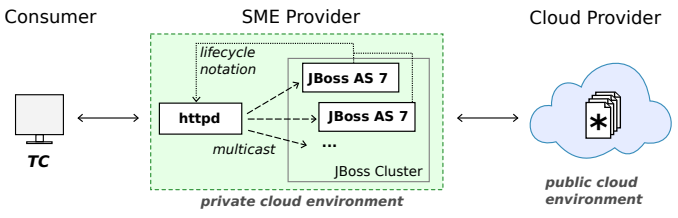
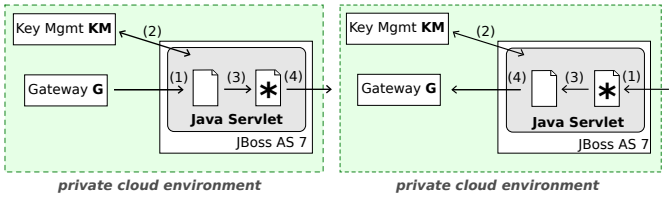
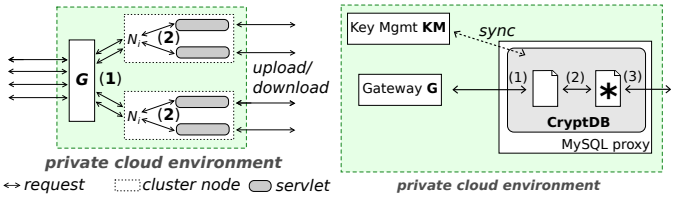


Figure 4. The internal JBoss Cluster inside the EL (green dashed box) for load balancing purposes. The figure shows the mod cluster enabled HTTP daemon (httpd) and the JBoss 7 application server nodes (JBoss AS 7), as well as the internal cluster communication to manage these JBoss Cluster nodes.



(a) file encryption (b) file decryption

Figure 5. Work and data flow inside the File Worker in the EL (green dashed box). Figures show encryption and decryption, respectively. The star symbolizes an encrypted file.



(a) Two Stage Scaling (b) CryptDB proxy

Figure 6. Two stage scaling while file processing inside the EL (green dashed box) and the work/data flow of the SQL query encryption.

3) Details of the Server Components: The server components of the encryption layer consist of a Java Servlet for file encryption and a MySQL proxy, based on the work of Popa et al. [33]. Figure 5(a) and Figure 5(b) show the procedure of file encryption and decryption, inside the JBoss AS7 Cluster nodes.

File encryption consists of four steps. First, receiving the HTTP request from the HTTP server (httpd) in gateway G (1). Second, checking if the file exists or creating a new key by sending a request to the key management system (Key Mgmt) (2), via a JDBC connection. Third, encrypting the file with a 256 bit encryption method (3). Fourth, upload the encrypted file to a file server in public cloud (4), using sardine, a WebDav client for Java [43].

File decryption is roughly the same, vice versa. First, downloading the encrypted file from the public cloud (1). Second, getting the encryption key from key management (2). Third, decrypting the file (3). Fourth, sending the file to the gateway or client, respectively (4).

Since the file up and downloads work in a synchronous way, a node would not be able to process another request, while uploading or downloading a file. This problem is naturally solved by JBoss AS 7 servers, by allowing many instances of the servlet in parallel, using threads. Actually, this is the second stage of scaling in the Encryption Layer. The two stages are shown in Figure 6(a). First a coarse-grained scaling in G, based on load balancing and virtual machines and cluster nodes (1) and second a fine-grained, based on parallel threads inside the nodes N_i (2). As mentioned above, there is of course the possibility of scaling the whole system, by having a redundant gateway, cluster and so on. This scenario is not considered, because of limitations of the available hardware for testing the prototype.

The servlet was developed in consideration of logging results and benchmarking different encryption methods. Table II shows test results of different encryption methods. The results show that AES (256Bit) works most efficient, considering encryption and decryption times. As a result of this and because of the fact AES is widely accepted as a secure encryption method, all following tests use AES (256Bit) for encryption. As mentioned, for detailed performance tests different times were logged. Because the File Workers FW_i nodes were not persistent, the logged times had to be sent to a logging service or fetched by one. Actually, we did not implement a separated logging service. We decided, to add the logged times in the HTTP header of the resulting HTTP response. Despite the small overhead this was very efficient for the prototype. Because of this approach all relevant performance measurement time bunched in the Test Client TC, the analysis become more easier. Logged data is:

1) start and communication times with key management
2) encryption method, and times spent for encryption and decryption
3) start and communication times with public cloud (upload, download, delete)

The second part of the server components is the integration of the MySQL proxy based CryptDB approach. The setup is described in *doc* folder in sources of the git repository of CryptDB [44] and consists basically of setting up some environment parameter and compiling an extended MySQL proxy. Our problem was to outsource the internal key management system of the CryptDB out of the SQL worker. Since the CryptDB approach uses an InnoDB database for internal consistency and backups, our attempt to forward these internal request to a remote database did not succeed. Therefore, we outsourced the database files in a very pragmatic way via NFS, leaving an improved solution open for future work. Another problem is scaling with databases. As far as our search results showed up, there is no comparable open source load balancer like mod_cluster available for databases. Actually, there are solutions like pgpool [45] and plproxy [46]. However, these tools are limited to a fixed pool of instances. Therefore, it is not possible to dynamically add some additional virtual machine or stop one VM. It is questionable if it really make sense to add additional database instances on demand, because principle like master-slave and database sharding are not suited for highly dynamic or short timed setups, which makes perfect sense for a persistence layer. Hence, existing solutions like Relational Cloud [47] go another way.

4) Details of the Test Client Implementation: The implementation of the Test Client TC is completely done in Java. TC has to fulfill three main tasks. First of all, the possibility to send HTTP file and SQL query requests to the EL. Second, the setup of different flexible test scenarios. And third, to measure and log the performance of the EL to analyze the test results. The first task is easily done by basic HTTP GET, POST and

TABLE II. UPLOAD, DOWNLOAD, ENCRYPTION, AND DECRYPTION AVERAGE TIMES $\bar{t}$ IN MILLISECONDS (FILE SIZE 1MB)

encryption method	$\bar{t}_{up}$[ms]	$\bar{t}_{enc}$[ms]	$\bar{t}_{down}$[ms]	$\bar{t}_{dec}$[ms]
AES (265 bit)	1040.9	39.9	1116.4	51.5
DESede (168 bit)	1165.9	167.2	1239.4	135.8
Serpent (256 bit)	1180.9	57.2	1138.2	57.9
Twofish (256 bit)	1195.9	50.6	1160.4	50.5
CAST6 (256 bit)	1300.9	53.3	1037.6	40.1

DELETE requests and a JDBC connection. These protocols are also used by the application server. For a direct communication with the WebDav file server in the public cloud again the sardine client is used. Actually, the CryptDB approach supports JDBC only in a very limited way. Nevertheless, it is enough for our EL prototype, to enable basic SQL statements for our tests. For the second task, we implement a highly configurable and effective possibility to create different load scenarios. Therefore, we can simulate autonomous clients with an individual behavior in parallel.

Figure 7 shows the basic parameters for configuration of the simulated clients. The figure illustrates a blue load line separated in *slack* and *spike* sectors. A slack sector represents a normal, average number of requests which cause a basic load to the EL. A spike sector represents a load peak, with a lot of requests in a short time span to cause some heavy load. In addition, Figure 7 shows a spike sector is always surrounded by two slack sectors. There can be an arbitrary number s of spikes, normally we set $1 \leq s \leq 3$ to finish the test scenarios in a reasonable time. As shown, there are a number of upload/download/delete blocks, called UDD. These blocks symbolize the procedure per file, which is at first uploaded to public cloud (through EL), after that download the file and check for identity. If successful the file is deleted in the cloud and UDD has processed successfully. This order is fixed, as one can not download or delete the file, until it is fully uploaded. The number n of UDD_n is configurable to $1 \leq n \leq 12$. In addition, the figure shows delays $D_{slack}(C_j, n_{slack}) = \Delta T(UDD_n, UDD_{n+1})$ and $D_{spike}(C_j, n_{spike}) = \Delta T(UDD_n, UDD_{n+1})$. Were $\Delta T(x, y)$ represents the timespan between the starting points of x and y, C_j is the client number j, and n_{slack}, n_{spike} are the numbers of UDD blocks in slack and spike, respectively. Normally, $D_{slack} \gg D_{spike}$, to simulate longer basic load periods and short peak load periods. The UDD blocks are independent, which means that up to n blocks can be processed in parallel. Last but not least, there is an option to choose the file size $F(C_j)$ between 1MB(F_1), 10MB(F_{10}) and 100MB(F_{100}). This file size is than fixed for this client C_j. Summing up, the configuration protocol for C_j works like processing the following protocol:

1) How many spikes s do you want?
2) How many UUD blocks should be sent in slack sector? (set n_{slack})
3) What is the (expected) delay? (D_{slack})
4) Choose a file size between F_1, F_{10} and F_{100}...
5) *// Repeat step 2 - 4 for spike configuration, then go to step 6*
6) Do you want to add another client C_{j+1}?

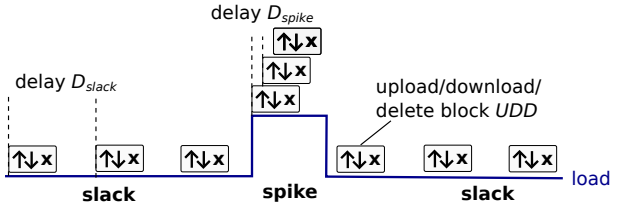


Figure 7. Client basic configuration for test scenarios with basic UDD blocks. One block consists of upload (↑), download(↓) and delete (x) of a file.

As question 6 suggests, there is the possibility to add another client. Basically, there is no limitation of the number of client we can set up. In case of adding another client, there will be a question how long the delay ΔT_{C_j} between the start of client simulation and start of C_j should be (simplification of $\Delta T(0, C_j)$). Slightly different is the setup of the SQL query client. In this case, we decided to implement a basic loop, consisting of a chosen number $N_{DML}(C_j)$ of data manipulations with INSERT, and DELETE statements, and a different number $N_{DQL}(C_j) \leq N_{DML}(C_j)$ of data queries with SELECT statements.

The third task, is done by logging the client configuration, the track of the configured protocol and by reading out the node-logs from the HTTP response headers. In addition, TC logs the status of the virtual machines of the private cloud environment. Therefore, the API of OpenNebula is used to log CPU and memory usage and in/outgoing network traffic for each VM. Because the OpenNebula System updates these values once every 20 seconds, the client fetches these VM performance measurements in the same period of time. Consequently, the logged data is:

1) start time of UDD blocks (+ details) and clients
2) VM CPU and memory usage, and in/outgoing network traffic
3) conglomeration of cluster node logs

The last important function TC provides is to run a comparison test. In this test setup the configured, simulated clients not only send their requests to G, but send them directly to the public cloud as well. This offers a good possibility to measure the overhead of the Encryption Layer.

5) Setup of the Public Cloud Servers: An Apache 2 HTTP server [48] and WebDav [49] is used to provide a file server in the public cloud. The file server is hosted by an European IaaS provider. For storing this data a common MySQL database extended by CryptDB user defined functions is also hosted by the IaaS provider. The extension is done by adding the CryptDB library *edb.so* to the plugins of the MySQL database and installing the NTL Library [50]. Both, virtual server use minimal resources of 1 GHz and 1 GB RAM.

Table III shows the summary of the setup of the Encryption Layer. We can see that the VMs have a decent usage of resources. Even if our hardware consists of standard components, this setup works very well for our EL prototype. The work of Toraldo [51] confirms that our setup in the private cloud is realistic.

B. Testing the prototype

The tests of the EL prototype are separated into load balancing and overhead test. A third scenario, testing dynamic scaling is not possible out of two reasons. Number one is discussed above, we called it image persistence dilemma. In our prototype setup it takes up to 5 min for a new VM to be ready to answer requests, including the time to transfer the 10 GB image and to boot up the VM. The second, more important point is that our available network bandwidth limits the load we can create in the Encryption Layer, making it unnecessary to improve our local image repository storage solution. This point is argued in the discussion section.

TABLE III. OVERVIEW OVER ENCRYPTION LAYER SETUP

Domain	VM name	vCPU [GHz]	RAM [GByte]	installed software
private Cloud	key management	0.50	0.25	MySQL database
	gateway	0.40	0.50	Apache 2, mod_cluster, MySQL proxy
	file worker	0.50	2.00	JBoss, Bouncycastle, Java servlet
	SQL worker	1.50	1.00	MySQL proxy, CryptDB
public cloud	file server	1.00	1.00	Apache 2, WebDAV
	SQL server	1.00	1.00	MySQL database, CryptDB MySQL module

1) Load balancing test scenario: Our aim is to show that client requests are equally distributed over two or more VMs in a dynamic way. Therefore, the JBoss cluster is configured in the above mentioned way and 2 File Worker VMs run in the private cloud. Figure 8 illustrates the protocol of the simulated clients by TC, sticking to the introduced symbolics above in Figure 7.

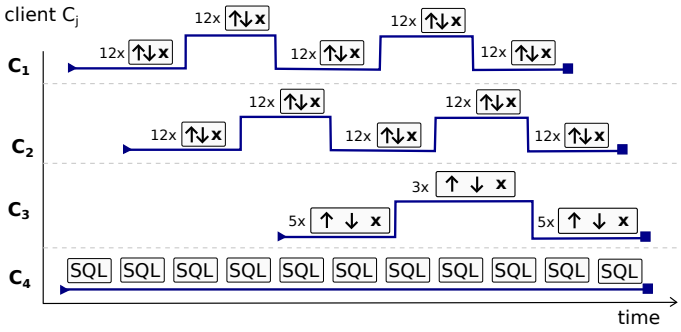


Figure 8. The illustration of the test protocol of the load balancing scenario. Four client are simulated to create load to the EL.

As shown, four clients are simulated, with the following parameters.

- $s_{C_{1,2}} = 2$, $s_{C_3} = 1$
- $\Delta T_{C_{1,4}} = 0$, $\Delta T_{C_2} = 20$ s, $\Delta T_{C_3} = 40$ s
- $D_{slack}(C_{1,2}, 12) = 2000$ ms
- $D_{spike}(C_{1,2}, 12) = 10$ ms
- $D_{slack}(C_3, 5) = 5000$ ms
- $D_{spike}(C_{1,2}, 3) = 10$ ms
- $F(C_{1,2}) = F_1$, $F(C_3) = F_{10}$
- $N_{DML}(C_4) = 15$, $N_{DQL}(C_4) = 10$

To complete the test protocol, the simulated clients take 10 min and 23 s.

2) Overhead test scenario: Our aim is to estimate the overhead of the EL. Therefore, a long running test has been set up. Figure 9 illustrates the protocol of the simulated client by TC.

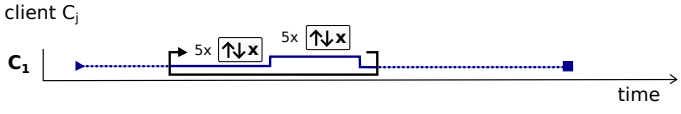


Figure 9. The test protocol for the long running overhead test scenario.

As shown, one clients is simulated, with the following parameters.

- $\Delta T_{C_1} = 0$,
- $D_{slack}(C_1, 5) = 20$ s, $D_{spike}(C_1, 5) = 20$ s
- $F(C_1) = F_1$

As Figure 9 and parameters show, this is a special setup with no peak load. In this scenario the spike symbolize the usage of our EL and the slack stands for a direct public cloud communication, without any encryption. Therefore, we can compare the results to get an estimation of the overhead. The test protocol is repeated until the user cancels it, in this case the test runs over 17 h.

Apart from these tests scenarios, we perform a lot of other tests to optimize parameters, like those of JBoss load balancing metrics, encryption methods and internal HTTP requests. We also run tests for SQL query processing comparison, by communicating directly with the database server in the public cloud. Despite of the given overhead values of CryptDB from Popa et al. [33], this enables us to estimate the overhead for SQL query encryption in our EL. This also allows to compare file encryption overheads with those from SQL query encryption.

C. Test results from the prototype

Like the test scenarios, the test results are separated in load balancing and overhead test results. The shown results are based on the data logged by TC, received from the JBoss cluster and the OpenNebula Management System. Figure 3 shows an abstract overview of the test setup. The detailed test scenarios are described in the section above.

1) Load Balancing Test Results: With the load balancing tests, we want to show how good the EL prototype can handle different load situations. The configured load balancing metric for the JBoss Cluster works very well. As mentioned, the metrics configured in mod_cluster config in JBoss nodes, combine CPU load, system memory usage and amount of outgoing/incoming requests traffic. Figure 10 shows the logged VM workloads of G and F_1, F_2 in 20 s intervals. The figure also show the number of client requests sent by TC in the specific interval. Please note that some client requests, like DELETE, do not cause much load. This is the reason, why the interval at 540 s has a low CPU load compared to the number of client requests. However, the timespan between 340 and 420 s is remarkable. A lot of upload and download requests were sent in this timespan. The gateway recognizes the high load of FW_2 sending client requests to FW_1. At time intervals of 380 s, it is inverse. In Figure 8 this is illustrated at the point when

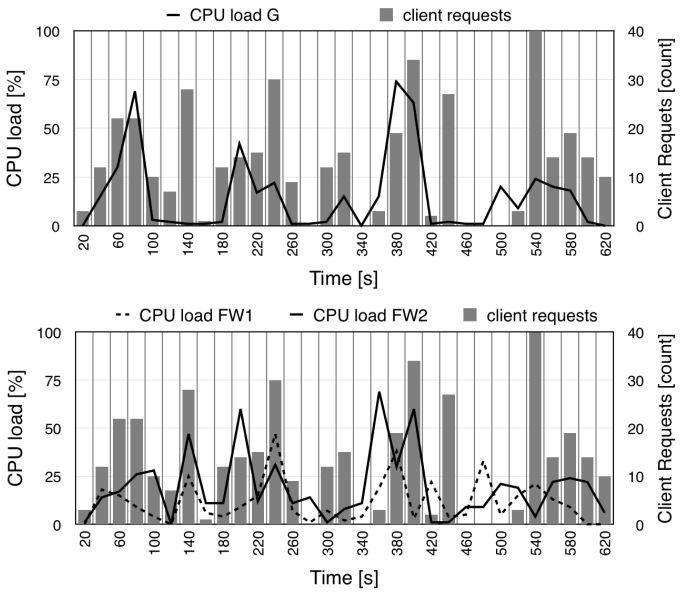
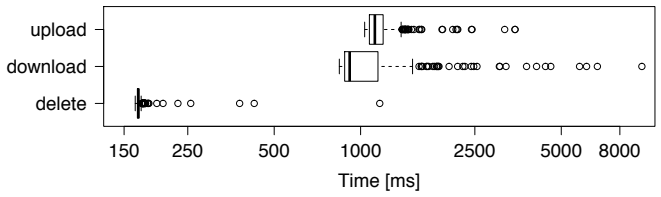
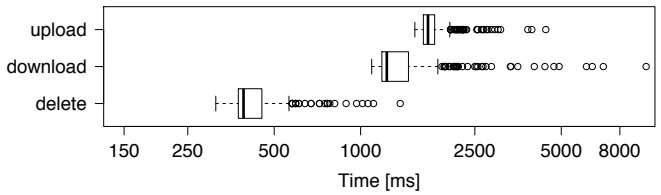


Figure 10. CPU load of G, FW_1, FW_2 VMs depending on the number of client requests.



(a) Direct cloud communication. Median times: Upload 1121 ms, Download 916 ms, Delete 162 ms



(b) Communication through Encryption Layer. Medians times: Upload 1717 ms, Download 1234 ms, Delete 391 ms

Figure 12. Logarithmic scaled response times for processing 515 uploads, 515 downloads and 515 delete file requests with and without the Encryption Layer. Whiskers maximums are 1.5 IQR

all clients $C_{1,2,3}$ have their spike loads at the same time. It has to be pointed out that Figure 8 is only a sketch, since clients C_i are completely independent and influenced by network latency and bandwidth. Figure 11 shows the response times for processing the SQL queries sent by TC though simulated client C_4 as illustrated in Figure 8.

2) Overhead Test Results: For the overhead test our aim is to get an estimation of a lower boundary of additional effort we have to accept, if we want to use on-the-fly encryption methods like an EL. Therefore, we ran some long term tests, to compare direct cloud communication with that through our EL. We uploaded, downloaded and deleted files in this test over 17 h. Figure 12(a) and 12(b) show the result box plots without and with the EL, respectively. Please note that the x-axis has a logarithmic scale. In Figure 12(a) we can see a very compact box plot for deleting files. This means the median of 168 ms for the deletion a file in the cloud storage is, except of very view outliers, nearly constant. Similar results can be seen in cases of upload and download the files. However, since these processes take much longer time, they are more influenced by a fluctuating network latency and bandwidth. Interestingly, the download is influenced much more. Figure 12(b) illustrates the box plots while using our EL. We can see the expected higher communication times. In addition, we can see that the data distribution in the box plot looks analogue to those in Figure 12(a). As we expected the delete time increases most. This is explainable because of the additional roundtrip to the key management system to delete the encryption key.

In another test we measured the response times of the database in cloud storage without our EL communication. Figure 13 illustrates the results. Please note that Figure 11 and Figure 13 have a different scale. Otherwise, the data is no longer readable, because of the high difference. Figure 11 shows results of the 18 rounds the SQL client C_4 executed its protocol. Remarkable in Figure 13 is that response times do have very few outliers. This is explainable in the short communication time and the short overall test time of around 6 min. In this short timespans the network latency fluctuations do not influence the results. Hence, the many outliers in Figure 11 do not result of a fluctuating network latency, since the duration of the test was only around 10 min. With regard to the logarithmic scale, a response times of ten to twenty seconds for a very basic query are unacceptable for practical use. Our best guess is the CryptDB proxy doing some internal recovery and key management operations. However, as mentioned we bind the relevant folder via NFS to our key management KM, some file operations via the internal network should not last that long. Figure 14 shows the percentage of times for processing a request through the EL. The figure splits up again in upload,

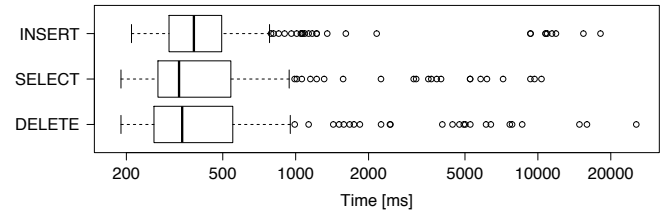


Figure 11. Logarithmic scaled response times for processing 234 INSERT, 180 SELECT and 234 DELETE queries through the Encryption Layer. Whiskers maximums are 1.5 IQR.

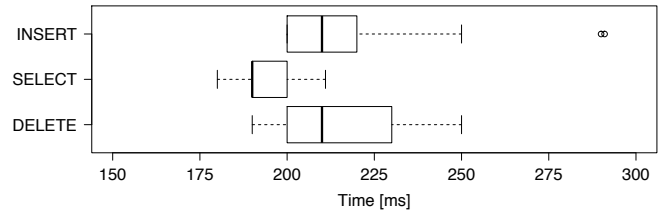


Figure 13. Response times for processing 310 INSERT, 340 SELECT and 310 DELETE queries direct to cloud. Whiskers maximums are 1.5 IQR.

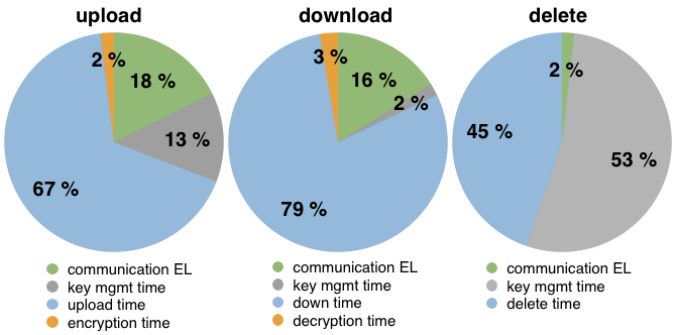


Figure 14. Percentage of times to encrypt and upload, decrypt and download, and delete a file, respectively.

download and delete parts. In cases of up- and download the communication time includes times spend for up- and download the file from TC to EL, therefore, the percentage is much higher than in case of deletion. More important is the fact that only two/three percent is used for encryption/decryption, the rest of time the File Workers are, roughly speaking, waiting to complete communications. Especially the communication overhead while deleting a file is significantly high, taking more than 50% of overall time.

The overheads for file encryption and query encryption can be seen in Figures 15 and 16. In fact, the figures illustrate the results of Figure 12, and Figures 11 and 13, respectively. In numbers, the overhead to upload and download a file, displayed in Figure 15 is around 53% and 34%, respectively. The overhead to delete a file is with around 132% very high. The difference can be explained by the required effort to store/fetch/delete the encryption keys and is also illustrated in Figure 14. As displayed in Figure 16 the overheads for query encryption are not as divers as the file encryption overheads. Nevertheless, their absolute values are higher, 81% for INSERT, 74% for SELECT and 62% for DELETE statements. It has to be pointed out, both Figures 15 and 16 display median, not average, times.

V. Discussion

We lead the discussion under the aspects of performance and overhead of the EL, and development and operation effort for SME provider. As mentioned, we used median times in our statistics. The reason is to estimate the lower boundary of additional effort for a scalable on-the-fly encryption system. Therefore, median times give a much better statement, because network or computation fluctuations do not influence that much. This is especially interesting to compare the SQL query encryption. As we figured out in our previous work [1], the current development status is highly prototypic and not usable for productive systems. Nevertheless, for an estimation we do need comparable values. If we assume a weighted average of 30% uploads/INSERTs, 60% downloads/SELECTs and 10% deletes, we get the following lower boundaries of average overhead $\overline{\Omega}$.

$$\overline{\Omega}_{fileEnc} = \frac{30 * 53 + 60 * 34 + 10 * 132}{100} = 49.5 \quad (2)$$

$$\overline{\Omega}_{queryEnc} = \frac{30 * 81 + 60 * 74 + 10 * 62}{100} = 74.9 \quad (3)$$

So, we got $\overline{\Omega}_{fileEnc}$ of around 50% and $\overline{\Omega}_{queryEnc}$ of around 75% for overall processing. We can see, that database encryption is more complicated than file encryption. This can be concluded from the fact that database encryption is not only a matter of data-at-rest encryption, but computation under encryption as well. However, the support of databases is limited in a way not all queries can be supported; details can be seen in Tu et al. [52].

As we can see in Figure 14 the main overheads results from communication with KM and EL. The time used for encryption and decryption is with two/tree percent very low. So, on the one hand, we agree with Huang and Xiaojiang [53], that protecting data with cryptography methods cause unavoidable overheads. However, on the other hand, we disagree with the statement, that it introduces heavy computational overhead. As we can see from Figure 14 the computational overhead is nearly negligible, most overhead results from network communication. So, network communication is the most influencing factor for the performance in our approach. Because the limiting network bandwidth of the internet access was the main reason we could not run scalability tests, since the computational effort for encryption/decryption was so low.

VI. Related work

Our approach applies typical security concepts in the field of cloud computing using tier, logic and data partitioning described by Bohli et al. [27]. In contrast to this study, we do not spread our tiers to various, non-collaborating cloud providers. We spread our solution over a private and public

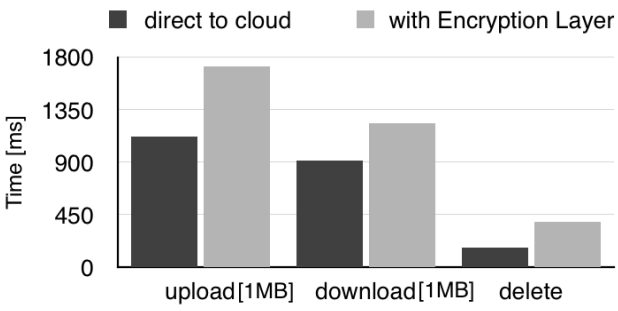


Figure 15. Diagram of median times to upload, download, and delete files with and without encryption

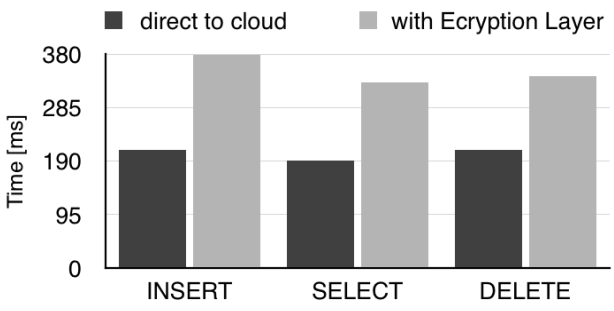


Figure 16. Diagram of median times to INSERT, SELECT, and DELETE queries with and without encryption

cloud, performing all critical tasks in the private cloud, like encryption and key management and outsource only the data tier in public cloud. This makes it unnecessary to label tasks or data as critical in a manual or semi-automatic manner, like described by Zhang et al. [21] and Oktay et al. [23]. Equally, Ray and Ganguly [22] present a data privacy model for cloud computing, in which sensitive and non-sensitive data is maintained separately. Zhou et al. [54] describe an evolved approach, using privacy-aware data retrieval, addressing problems of anonymization like quasi identifiers. Wang et al. [16] give a short overview over hybrid cloud security. Their *Single Encryption* scenario is in that way similar to ours that data is encrypted in private cloud before transferring in public cloud. However, their focus lay on authentication model for inter (multi-) cloud communication. A data management method, via a data portfolio, for company data in a hybrid cloud configuration, is discussed by Tanimoto et al. [24]. Li et al. [28] describe a framework for SME to orchestrate cloud services in multi-cloud environments. Their work tries to integrate applications of the internal information system of SME with a public e-business platform.

VII. Future work

An appropriate prototype of the discussed hybrid key management system is one of the very next steps in future work, in order to have a fully functional hybrid cloud environment. In addition, we could not run scaling tests because of limited hardware resources, we leave this open for future work. Our test results confirm the mentioned fact by Popa et al. [33] that the implementation of CryptDB is highly prototypical. As the own implementations of Google and SAP show (for details see [55]) more development effort, e.g., towards full support of JDBC - is necessary. Figure 14 points out that the encryption workload of file worker is not high. This leaves space for additional functionalities like file compressing, for faster up- and downloads, or file indexing for possible searches over the encrypted files. The latter is mentioned in [56]. Also, the integration of a secure identity and key management system, e.g., Kerberos [57], is required to provide a SaaS solution with focus on the customers privacy. Moreover, the tests show that implementations of failure and backup routines are absolutely necessary. Despite the different focus in this first implementation, we want to point out that security is not only about protecting data from unauthorized access or viewing, but also issues of auditing, data-integrity, and reliability should be concerned too.

VIII. Conclusion

In this this paper the introduction of an additional architecture layer, called Encryption Layer (EL) is described. With this layer SME can address privacy and security issues through the usage of encryption methods. SME have the option to basically employ three cloud models (private, public and hybrid) as their solution, which were discussed and compared in Section II. The private cloud model is the most inflexible and cost intensive option. Probably being an option for large companies owning much hardware resources, it is not suitable for SME. The public cloud model is the preferred choice for SME if the application has no particular high privacy or security requirements. Our work, which is focused on a hybrid cloud concept, offers a compromise between security/privacy, efficiency and costs. Therefore, we describe our developed prototype of the EL in detail. The prototype includes scalable encryption server for files and queries, and a basic key management system inside a private cloud environment based on OpenNebula 4.4. In addition, we set up file and database server in a public cloud environment. Problems as the so called image persistence dilemma were discussed and appropriate solutions are suggested. Moreover, we proposed a hybrid key management system, which can be used to distribute key management task between consumer and SME in order to achieve a higher privacy for service customers. Our test results showed that load balancing inside the private cloud, using a JBoss Cluster works well. Besides, we showed the main percentage of the overheads of 50% - 75% is a matter of communication, and not of heavy computation because of encryption and decryption. Therefore, network communication is the most influencing factor for performance. To be applicable in productive systems, improvements of performance and more research are necessary. Nevertheless, we show that SME can establish solutions like our EL to solve security and privacy issues with a reasonable amount of effort. Our work provides a good basic solution for SME to get their first experiences in the cloud computing business.

Acknowledgment

The published work has been financially supported by the European Social Fund (ESF) and the EU. We would like to thank the anonymous reviewers for helpful comments.

References

[1] P. Reinhold, W. Benn, B. Krause, F. Goetz, and D. Labudde, “Hybrid cloud architecture for software-as-a-service provider to achieve higher privacy and decrease security concerns about cloud computing,” in CLOUD COMPUTING 2014, The Fifth International Conference on Cloud Computing, GRIDs, and Virtualization, 2014, pp. 94–99.

[2] Instagram. Website. [Online]. Available: http://instagram.com [retrieved: November, 2014]

[3] Dropbox. Website. [Online]. Available: https://www.dropbox.com [retrieved: November, 2014]

[4] F. Gens, “It cloud services user survey, pt. 2: Top benefits & challenges,” IDC eXchange, 2008.

[5] ——, “New idc it cloud services survey: top benefits and challenges,” IDC exchange, 2009.

[6] BITOM and KMPG, “Cloud-monitor 2013 cloud-computing in deutschland – status quo und perspektiven,” KMPG Study, February 2013.

[7] CrispResearch. Study platform-as-a-service: German sme market survey. [Online]. Available: http://www.business-cloud.de/wp-content/uploads/2014/07/STUDIE-Platform-as-a-Service01.pdf [retrieved: November, 2014]

[8] F. Liu, J. Tong, J. Mao, R. Bohn, J. Messina, L. Badger, and D. Leaf, “Nist cloud computing reference architecture,” Tech. Rep., 2011.

[9] P. Mell and T. Grance, “The nist defintion of cloud computing,” National Institute of Standards and Technology, Tech. Rep., 2011.

[10] E. Aguiar, Y. Zhang, and M. Blanton, “An overview of issues and recent developments in cloud computing and storage security,” in High Performance Cloud Auditing and Applications. Springer, 2014, pp. 3–33.

[11] S. Jie, J. Yao, and C. Wu, “Cloud computing and its key techniques,” in Electronic and Mechanical Engineering and Information Technology (EMEIT), 2011 International Conference on, vol. 1. IEEE, 2011, pp. 320–324.

[12] S. Subashini and V. Kavitha, "A survey on security issues in service delivery models of cloud computing," Journal of Network and Computer Applications, vol. 34, no. 1, 2011, pp. 1–11.

[13] J. Rhoton and R. Haukioja, Cloud Computing Architected: Solution Design Handbook, 2013th ed. Recursive Press, 2013.

[14] D. A. Fernandes, L. F. Soares, J. V. Gomes, M. M. Freire, and P. R. Inácio, "Security issues in cloud environments: a survey," International Journal of Information Security, vol. 13, no. 2, 2014, pp. 113–170.

[15] S. Van Hoecke, T. Waterbley, J. Devos, T. Deneut, and J. De Gelas, "Efficient management of hybrid clouds," in CLOUD COMPUTING 2011, The Second International Conference on Cloud Computing, GRIDs, and Virtualization, 2011, pp. 167–172.

[16] J. K. Wang and X. Jia, "Data security and authentication in hybrid cloud computing model," in Global High Tech Congress on Electronics (GHTCE). IEEE, 2012, pp. 117–120.

[17] M. A. Rahaman. How secure is sap business bydesign for your business. [Online]. Available: http://scn.sap.com/docs/DOC-26472 [retrieved: November, 2014]

[18] C. Gentry, "A Fully Homomorphic Encryotion Scheme," Ph.D. dissertation, Stanford University, 2009.

[19] R. A. Popa, E. Stark, J. Helfer, S. Valdez, N. Zeldovich, M. F. Kaashoek, and H. Balakrishnan, "Building web applications on top of encrypted data using mylar," in USENIX Symposium of Networked Systems Design and Implementation, 2014.

[20] M. van Dijk, A. Juels, A. Oprea, R. L. Rivest, E. Stefanov, and N. Triandopoulos, "Hourglass schemes: how to prove that cloud files are encrypted," in Proceedings of the 2012 ACM conference on Computer and communications security. ACM, 2012, pp. 265–280.

[21] K. Zhang, X. Zhou, Y. Chen, and X. Wang, "Sedic : Privacy-Aware Data Intensive Computing on Hybrid Clouds Categories and Subject Descriptors," 2011, pp. 515–525.

[22] C. Ray and U. Ganguly, "An approach for data privacy in hybrid cloud environment," in Computer and Communication Technology (ICCCT), 2011 2nd International Conference on. IEEE, 2011, pp. 316–320.

[23] K. Y. Oktay, V. Khadilkar, B. Hore, M. Kantarcioglu, S. Mehrotra, and B. Thuraisingham, "Risk-aware workload distribution in hybrid clouds," in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. IEEE, 2012, pp. 229–236.

[24] S. Tanimoto, Y. Sakurada, Y. Seki, M. Iwashita, S. Matsui, H. Sato, and A. Kanai, "A study of data management in hybrid cloud configuration," in Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2013 14th ACIS International Conference on. IEEE, 2013, pp. 381–386.

[25] Amazon. Aws s3. [Online]. Available: http://aws.amazon.com/s3/ [retrieved: November, 2014]

[26] Microsoft. Azure sql database. [Online]. Available: http://azure.microsoft.com/en-us/services/sql-database/ [retrieved: November, 2014]

[27] J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," Dependable and Secure Computing, IEEE Transactions on, vol. 10, no. 4, 2013, pp. 212–224.

[28] Q. Li, Z.-y. Wang, W.-h. Li, J. Li, C. Wang, and R.-y. Du, "Applications integration in a hybrid cloud computing environment: modelling and platform," Enterprise Information Systems, vol. 7, no. 3, 2013, pp. 237–271.

[29] M. Hussain, "The Design and Applications of a Privacy-Preserving Identity and Trust-Management System," Ph.D. dissertation, Queen's University (Kingston, Ont.), 2010.

[30] S. Zarandioon, D. D. Yao, and V. Ganapathy, "K2c: Cryptographic cloud storage with lazy revocation and anonymous access," in Security and Privacy in Communication Networks. Springer, 2012, pp. 59–76.

[31] Apple. ios security. [Online]. Available: http://images.apple.com/iphone/business/docs/iOS_Security_Feb14.pdf [retrieved: November, 2014]

[32] Quarkslab. imessage privacy. [Online]. Available: http://blog.quarkslab.com/static/resources/2013-10-17_imessage-privacy/slides/iMessage_privacy.pdf [retrieved: November, 2014]

[33] R. A. Popa, C. Redfield, N. Zeldovich, and H. Balakrishnan, "Cryptdb: protecting confidentiality with encrypted query processing," in Proceedings of the Twenty-Third ACM Symposium on Operating Systems Principles. ACM, 2011, pp. 85–100.

[34] OpenNebula. Website. [Online]. Available: http://docs.opennebula.org/4.4/release_notes/ [retrieved: November, 2014]

[35] QEMU. Qcow3. [Online]. Available: http://wiki.qemu.org/Features/Qcow3 [retrieved: November, 2014]

[36] Sonartype. Nexus. [Online]. Available: http://www.sonatype.org/nexus/ [retrieved: November, 2014]

[37] ONEFlow. Documentation. [Online]. Available: http://docs.opennebula.org/4.4/advanced_administration/application_flow_and_auto-scaling/appflow_configure.html [retrieved: November, 2014]

[38] F. Marchioni, JBoss AS 7 Configuration, Deployment and Administration. Packt Publishing Ltd, 2011.

[39] ModCluster. mod-cluster website. [Online]. Available: http://www.jboss.org/mod_cluster [retrieved: November, 2014]

[40] ——. Documentation. [Online]. Available: http://docs.jboss.org/mod_cluster/1.2.0/html/ [retrieved: November, 2014]

[41] Redhat. Mod cluster subsystem. [Online]. Available: https://access.redhat.com/documentation/en-US/JBoss_Enterprise_Application_Platform/6.1/html/Administration_and_Configuration_Guide/Configure_the_mod_cluster_Subsystem.html [retrieved: November, 2014]

[42] Bouncycastle. Website. [Online]. Available: http://www.bouncycastle.org/java.html [retrieved: November, 2014]

[43] G. R. Sardine. Sardine - an easy to use webdav client for java. [Online]. Available: https://github.com/lookfirst/sardine [retrieved: November, 2014]

[44] Git-Repository. cryptdb. [Online]. Available: git://g.csail.mit.edu/cryptdb [retrieved: November, 2014]

[45] Pgpool. Website. [Online]. Available: http://www.pgpool.net/mediawiki/index.php/Main_Page [retrieved: November, 2014]

[46] Plproxy. Website. [Online]. Available: http://plproxy.projects.pgfoundry.org/doc/tutorial.html [retrieved: November, 2014]

[47] C. Curino, E. P. C. Jones, R. A. Popa, N. Malviya, E. Wu, S. Madden, H. Balakrishnan, and N. Zeldovich, "Relational cloud: A database-as-a-service for the cloud," in 5th Biennial Conference on Innovative Data Systems Research, CIDR, 2011.

[48] Apache. Apache 2.2 website. [Online]. Available: http://httpd.apache.org/docs/2.2/en/ [retrieved: November, 2014]

[49] WebDAV. Website. [Online]. Available: http://www.webdav.org [retrieved: November, 2014]

[50] NTL-Library. Ntl: A library for doing number theory. [Online]. Available: http://www.shoup.net/ntl/ [retrieved: November, 2014]

[51] G. Toraldo, OpenNebula 3 Cloud Computing. Packt Publishing Ltd, 2012.

[52] S. Tu, M. F. Kaashoek, S. Madden, and N. Zeldovich, "Processing analytical queries over encrypted data," in Proceedings of the 39th international conference on Very Large Data Bases. VLDB Endowment, 2013, pp. 289–300.

[53] X. Huang and X. Du, "Efficiently secure data privacy on hybrid cloud," in Communications (ICC), 2013 IEEE International Conference on. IEEE, 2013, pp. 1936–1940.

[54] Z. Zhou, H. Zhang, X. Du, P. Li, and X. Yu, "Prometheus: Privacy-aware data retrieval on hybrid cloud," in INFOCOM, 2013 Proceedings IEEE, 2013, pp. 2643–2651.

[55] CryptDB. Website impact section. [Online]. Available: http://css.csail.mit.edu/cryptdb/#Impact [retrieved: November, 2014]

[56] S. Kamara, C. Papamanthou, and T. Roeder, "Cs2: A searchable cryptographic cloud storage system," Microsoft Research, TechReport MSR-TR-2011-58, 2011.

[57] Kerberos. Keberos documentation. [Online]. Available: http://tools.ietf.org/html/rfc4120 [retrieved: November, 2014]

Evaluating Parallel Breadth-First Search Algorithms for Multiprocessor Systems

Matthias Makulla and Rudolf Berrendorf
Computer Science Department
Bonn-Rhein-Sieg University
Sankt Augustin, Germany
e-mail: matthias.makulla@h-brs.de, rudolf.berrendorf@h-brs.de

***Abstract*—Breadth-First Search is a graph traversal technique used in many applications as a building block, e.g., to systematically explore a search space or to determine single source shortest paths in unweighted graphs. For modern multicore processors and as application graphs get larger, well-performing parallel algorithms are favorable. In this paper, we systematically evaluate an important class of parallel algorithms for this problem and discuss programming optimization techniques for their implementation on parallel systems with shared memory. We concentrate our discussion on level-synchronous algorithms for larger multicore and multiprocessor systems. In our results, we show that for small core counts many of these algorithms show rather similar performance behavior. But, for large core counts and large graphs, there are considerable differences in performance and scalability influenced by several factors, including graph topology. This paper gives advice, which algorithm should be used under which circumstances.**

***Index Terms*—parallel breadth-first search, BFS, NUMA, memory bandwidth, data locality.**

I. Introduction

Breadth-First Search (BFS) is a visiting strategy for all vertices of a graph. BFS is most often used as a building block for many other graph algorithms, including shortest paths, connected components, bipartite graphs, maximum flow, and others [1] [2] [3]. Additionally, BFS is used in many application areas where certain application aspects are modeled by a graph that needs to be traversed according to the BFS visiting pattern. Amongst others, exploring state space in model checking, image processing, investigations of social and semantic graphs, machine learning are such application areas [4].

We are interested in undirected graphs $G = (V,E)$, where $V = \{v_1,...,v_n\}$ is a set of vertices and $E = \{e_1,...,e_m\}$ is a set of edges. An edge e is given by an unordered pair $e = (v_i,v_j)$ with $v_i,v_j \in V$. The number of vertices of a graph will be denoted by $|V| = n$ and the number of edges is $|E| = m$.

Assume a connected graph and a source vertex $v_0 \in V$. For each vertex $u \in V$ define $depth(u)$ as the number of edges on the shortest path from v_0 to u, i.e., the edge distance from v_0. With $depth(G)$ we denote the depth of a graph G defined as the maximum depth of any vertex in the graph *relative to the given source vertex*. Please be aware that this may be different to the diameter of a graph, the largest distance between *any* two vertices.

The problem of BFS for a given graph $G = (V,E)$ and a source vertex $v_0 \in V$ is to visit each vertex in a way such that a vertex v_1 must be visited before any vertex v_2 with $depth(v_1) < depth(v_2)$. As a result of a BFS traversal, either the level of each vertex is determined or a (non-unique) BFS spanning tree with a father-linkage of each vertex is created. Both variants can be handled by BFS algorithms with small modifications and without extra computational effort. The problem can be easily extended and handled with directed or unconnected graphs. A sequential solution to the problem can be found in textbooks based on a queue where all non-visited adjacent vertices of a visited vertex are enqueued [2] [3] [5]. The computational complexity is $O(|V|+|E|)$. Level-synchronous BFS algorithms work in parallel on all vertices of one level and have a barrier synchronization [6] before the work for the next level is launched.

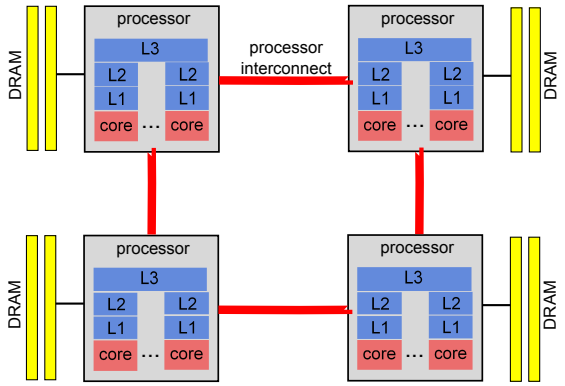


Fig. 1: Principial Structure of a 4-socket Multiprocessor Multicore NUMA system.

Large parallel systems with shared memory are nowadays organized as multiprocessor multicore system in a Non-Uniform Memory Access topology [7] (NUMA; see Fig. 1). In such systems multiple processors (usually with multiple cores) are connected by a fast interconnection network, e.g., QuickPath Interconnect (QPI) on Intel systems [8] or HyperTransport (HT) on AMD systems [9]. All processors / cores share a common address space in a DRAM based main memory. The interesting aspect is that this main memory / address space is distributed to the NUMA nodes. This has consequences for the performance aware programmer as accessing data on processor i that resides on DRAM chips that are assigned / close to processor i is faster than accessing data residing in DRAM chips that are assigned to a different / further away processor. Additionally, the coherence protocol may invalidate cached data in the cache of one processor because another processor modifies the same data. As a consequence, for a programmer

often a global view on parallel data structures is necessary to circumvent performance degradation related to coherence issues.

Many parallel BFS algorithms got published (see Section II for a comprehensive overview including references), all with certain scenarios in mind, e.g., large distributed memory parallel systems using the message passing programming model [10] [11] [12], algorithms variants that are tailored to Graphic Processing Units (GPU) using a different parallel programming model [13] [14] [15], or randomized algorithms for fast, but possibly sub-optimal results [16]. Such original work often contains performance data for the newly published algorithm on a certain system, but often just for the new approach, or taking only some parameters in the design space into account [17] [18]. To the best of our knowledge, there is no rigid comparison that systematically evaluates relevant parallel BFS algorithms in detail in the design space with respect to parameters that may influence the performance and/or scalability and give advice, which algorithm is best suited for which application scenario. In this paper, BFS algorithms of a class with a large practical impact (level-synchronous algorithms for shared memory parallel systems) are systematically compared to each other.

The paper first gives an overview on parallel BFS algorithms and classifies them. Second, and this is the main contribution of the paper, a selection of level-synchronous algorithms relevant for the important class of multicore and multiprocessors systems with shared memory are systematically evaluated with respect to performance and scalability. The results show that there are significant differences between algorithms for certain constellations, mainly influenced by graph properties and the number of processors / cores used. No single algorithm performs best in all situations. We give advice under which circumstances which algorithms are favorable.

The paper is structured as follows. Section II gives a comprehensive overview on parallel BFS algorithms with an emphasis on level synchronous algorithms for shared memory systems. Section III prescribes algorithms in detail that are of concern in this paper. Section IV describes our experimental setup, and, in Section V, the evaluation results are discussed, followed by a conclusion.

II. Related Work and Parallel BFS Algorithms

We combine in our BFS implementations presented later in Section III several existing algorithmic approaches and optimization techniques. Therefore, the presentation of related work has to be intermingled with an overview on parallel BFS algorithms itself.

In the design of a parallel BFS algorithm different challenges might be encountered. As the computational density for BFS is rather low, BFS is memory bandwidth limited for large graphs and therefore bandwidth has to be handled with care. Additionally, memory accesses and work distribution are both irregular and dependent on the data / graph topology. Therefore, in large NUMA systems data layout and memory access should respect processor locality [19]. In multicore multiprocessor systems, things get even more complicated, as several cores share higher level caches and NUMA-node memory, but have distinct and private lower-level caches (see Fig. 1 for an illustration).

A more general problem for many parallel algorithms including BFS is a sufficient load balance of work to parallel threads when static partitioning is not sufficient, e.g., distributing statically all vertices in blocks over available cores. Even if an appropriate mechanism for load balancing is deployed, graphs might only supply a limited amount of parallelism. As the governing factor that influences workload and parallelism is the average vertex degree of the traversed graph, especially graphs with a very low average vertex degree are challenging to most algorithms. This aspect notably affects the popular level-synchronous approaches for parallel BFS we concentrate on later. We discuss this aspect in Section V.

The output of an BFS algorithm is an array `level` of size n that stores in `level[v]` the level found for the vertex v. This array can be (mis-)used to keep track of unvisited vertices, too, e.g., initially storing the value -1 in all array elements marking all vertices as unvisited. We discuss in Section II-D other possibilities. As discussed, in BFS algorithms housekeeping has to be done on visited / unvisited vertices as well as frontiers with several possibilities how to do that. A rough classification of algorithms can be achieved by looking at these strategies. Some of them are based on special container structures where information has to be inserted and deleted. Scalability and administrative overhead of these containers are of interest. Many algorithms can be classified into two groups: *container centric* and *vertex centric* approaches.

Important for level-synchronous algorithms is the notion of a level and correlated to that a (vertex) frontier. Fig. 2 explains that notion on an example graph and three level iterations of a level-synchronous BFS algorithm. Starting with a given vertex v_0 this vertex makes up the initial vertex frontier and gets assigned the level 0. All unvisited neighbors of all vertices of the current frontier are part of the next frontier and get the current level plus 1. Such a level iteration is repeated until no more unvisited vertices exist. As can be seen for the example graph in Fig. 2, housekeeping has to be done whether a vertex is visited or not. And working on several vertices of the same level in parallel may lead to a situation where several threads may detect in parallel that a vertex is unvisited. Therefore, care has to be taken to handle such situations, e.g., with synchronization or handling unsynchronized accesses in a appropriate way [20]. As explained before, we concentrate our discussion on connected graphs. To extend BFS for graphs that are not connected, another BFS traversal can be started if one BFS traversal stops with any then unvisited vertex as long as unvisited vertices exist.

A. Container Centric Approaches

The emphasis in this paper is on level-synchronous algorithms where data structures are used, which store the current and the next vertex frontier. Generally speaking, these approaches deploy two identical containers (*current* and *next*)

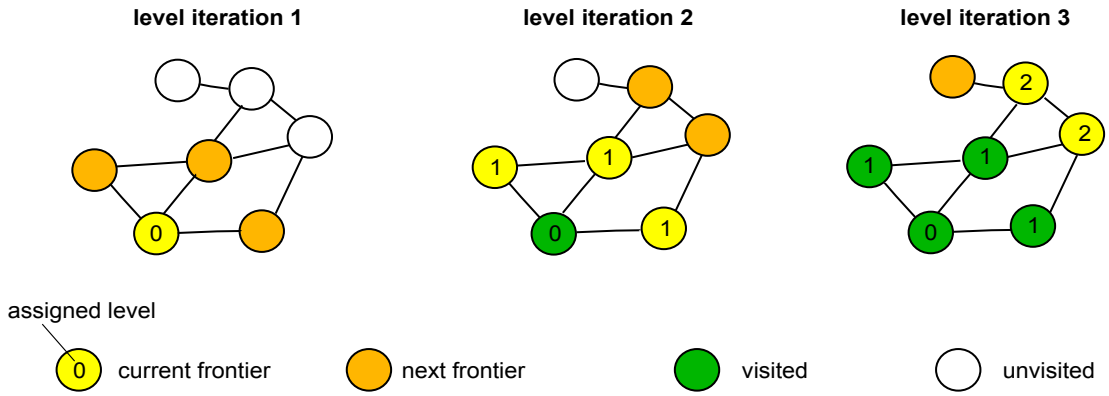


Fig. 2: Vertex levels and frontiers.

whose roles are swapped at the end of each iteration. Usually, each container is accessed in a concurring manner such that the handling/avoidance of synchronized accesses becomes crucial. Container centric approaches are eligible for dynamic load balancing but are sensible to data locality on NUMA systems. Container centric approaches for BFS can be found in some parallel graph libraries [21] [22] [23] [24].

For level synchronous approaches, a simple list (for example an array based list) is a sufficient container. There are approaches, in which each thread manages two private lists to store the vertex frontiers and uses additional lists as buffers for communication [10] [25] [26]. Each vertex is associated with a certain thread, which leads to a static one dimensional partitioning of the graph's vertices. When one thread encounters a vertex associated to another thread while processing the adjacent vertices of its local vertex frontier, it adds this foreign vertex to the communication buffer of the owning thread. After a barrier synchronization each thread processes the vertices contained in the communication buffers of each foreign thread. As the number of threads increases, an increased number of communication buffers must be allocated, limiting the scalability of this approach. Due to the one dimensional partitioning data locality can be utilized.

In contrast this approach completely neglects load balancing mechanisms. The very reverse would be an algorithm, which focuses on load balancing. This can be achieved by using special lists that allow concurrent access of multiple threads. In contrast to the thread private lists of the previous approach, two global lists are used to store the vertex frontiers. The threads then concurrently work on these lists and implicit load balancing can be achieved. Concurrent lock-free lists can be efficiently implemented with an atomic compare-and-swap operation.

It is possible to combine both previous approaches and create a well optimized method for NUMA architectures [17] [18]. While a global list for the current vertex frontier supplies fundamental work balance, it completely ignores data locality. In the NUMA optimized approach each vertex is assigned to one memory bank, leading to a one dimensional vertex partitioning among the sockets / NUMA nodes. As described above communication buffers are managed for each socket that gather foreign vertices. The local vertex frontier for one socket is a concurrent list, which is processed in parallel by the threads belonging to this socket.

Furthermore, lists can be utilized to implement container centric approaches on special hardware platforms as graphic accelerators with warp centric programming [13] [27]. Instead of threads, warps (groups of threads; [15]) become the governing parallel entities. With warp centric programming, BFS is divided into two phases: SISD and SIMD. In a SISD phase, each thread of a warp executes the same code on the same data. These phases are used to copy global vertex chunks to warp local memory. Special hardware techniques support efficient copying on warp level. In the SIMD phase each thread executes the same statements but on different data. This is used to process the adjacent vertices of one vertex in parallel. Because the workload is split into chunks and gradually processed by the warps, fundamental work balancing is ensured. To avoid ill-sized chunks another optimization may be applied: deferring outliers [13], which will be discussed in a later section.

Besides strict FIFO (First-In-First-Out) and relaxed list data structures, other specialized containers may be used. A notable example is the *bag* data structure [28] [29], which is optimized for a recursive, task parallel formulation of a parallel BFS algorithm. This data structure allows an elegant, object-oriented implementation with implicit dynamic load balancing, but which regrettably lacks data locality or rather leaves it solely to a thread runtime system. A *bag* is an array of a special kind of binary trees that holds the vertices of the current and the next vertex frontier. One can insert new nodes into a *bag*, split one *bag* into two almost equal sized *bags* and unify two *bags* to form a new *bag*. All operations are designed to work with minimal complexity. When beginning a new BFS level iteration one *bag* forms the entire vertex frontier for this iteration. The initial *bag* is then split into two parts. The first part is used for the current thread and the second is used to spawn a new thread (or parallel task). A thread splits and spawns new tasks until its input *bag* is smaller than a specified threshold. In this case the thread processes the vertices from

its partial *bag* and inserts the next frontier's vertices into a new *bag*. When two threads finish processing their *bags*, both results are unified. This is recursively done until only one *bag* remains, which then forms the new vertex frontier.

B. Vertex Centric Approaches

A vertex centric approach achieves parallelism by assigning a parallel entity (e.g., a thread) to each vertex of the graph. Subsequently, an algorithm repeatedly iterates over all vertices of the graph. As each vertex is mapped to a parallel entity, this iteration can be parallelized. When processing a vertex, its neighbors are inspected and if unvisited, marked as part of the next vertex frontier. The worst case complexity for this approach is therefore $O(n^2)$ for degenerated graphs (e.g., linear lists). This vertex centric approach might work well only, if the graph depth is very low.

A vertex centric approach does not need any additional data structure beside the graph itself and the resulting $level$-/$father$-array that is often used to keep track of visited vertices. Besides barrier synchronization at the end of a level iteration, a vertex centric approach does with some care not need any additional synchronization. The implementation is therefore rather simple and straightforward. The disadvantages of vertex centric approaches are the lacking mechanisms for load balancing and graphs with a large depth.

But this overall approach makes it well-suited for GPU's where each vertex is mapped to exactly one thread [30] [31]. This approach can be optimized further by using hierarchical vertex frontiers to utilize the memory hierarchy of a graphic accelerator, and by using hierarchical thread alignment to reduce the overhead caused by frequent kernel restarts [32].

Their linear memory access and the possibility to take care of data locality allow vertex centric approaches to be efficiently implemented on NUMA machines [27]. Combined with a proper partitioning, they are also suitable for distributed systems, as the overhead in communication is rather low. But as pointed out already above, this general approach is suited only for graphs with a very low depth.

C. Other Approaches

The discussion in this paper concentrates on level-synchronous parallel BFS algorithms for shared-memory parallelism. There are parallel algorithms published that use different approaches or that are designed for other parallel architectures in mind. In [16], a probabilistic algorithm is shown that finds a BFS tree with high probability and that works in practice well even with high-diameter graphs. Beamer et al. [33] combines a level-synchronous top-down approach with a vertex-oriented bottom-up approach where a heuristic switches between the two alternatives; this algorithm shows for small world graphs very good performance. Yasui et al. [34] explores this approach in more detail for multicore systems. In [35], a fast GPU algorithm is introduced that combines fast primitive operations like prefix sums available with highly-optimized libraries. A task-based approach for a combination of CPU/ GPU is presented by Munguia et al. [36].

A BFS traversal can be implemented as a matrix-vector product over a special semi-ring [37]. Matrix-vector multiplication is subject to elaborated research and one could profit from highly optimized parallel implementations when implementing BFS as a matrix-vector product.

Additionally, there are (early) algorithms that focus more on a principial approach while ignoring important aspects of real parallel systems [38] [39].

For distributed memory systems, the partitioning of the graph is crucial. Basically, the two main strategies are one dimensional partitioning of the vertices and two dimensional edge partitioning [10]. The first approach is suited for small distributed and most shared memory systems, while the second one is viable for large distributed systems. Optimizations of these approaches combine threads and processes in a hybrid environment [37] and use asynchronous communication [40] to tolerate communication latencies: one dedicated communication thread per process is used to take care of all communication between the different processes. The worker threads communicate via multiple producer / single consumer queues with the communication thread [40]. Each worker writes the non-local vertices to the proper queue while the communication thread monitors these queues. When a queue is full, the communicator sends its contents to the associated process. This reduces the communication overhead at the end of an iteration. Scarpazza discusses in [41] optimizations for the Cell/B.E. processor. Pearce [42] discusses techniques to use even NAND based Flash memories for very large graphs that do not fit into main memory.

D. Common extensions and optimizations

An optimization applicable to some algorithms is the use of a bitmap to keep track of visited vertices in a previous iteration [17] instead of (mis-)using the level information to keep track of unvisited vertices (e.g., `level[v]` equals -1 for an unvisited vertex *v*). The intention is to keep more information on visited vertices in a higher level of the cache hierarchy as well as to reduce memory bandwidth demands. Bitmaps can be used to optimize container as well as vertex centric approaches.

Fine-grained tuning like memory pre-fetching can be used to tackle latency problems [18] (but which might produce even more pressure on memory bandwidth).

Most container centric approaches work on vertex chunks instead of single vertices [28]. This reduces container access and synchronization overhead.

To avoid unequal workloads on different threads another optimization is to defer outliers [13] that could slow down one thread and force all others to go idle and wait for the busy thread at the end of an iteration. Usually, outliers are vertices with an unusual high vertex degree. When a thread encounters an outlying vertex, this vertex is inserted into a special container. This way processing outliers is deferred until the end of an iteration, avoiding unequal distribution of workload among the threads.

Besides implicit load balancing of some container centric approaches, there exist additional methods. One is based on

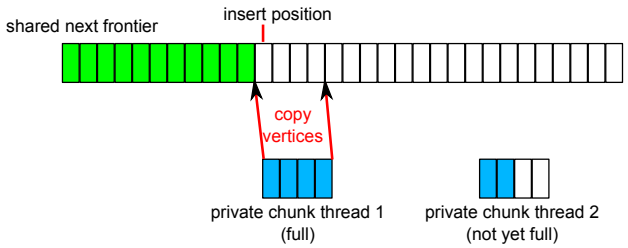


Fig. 3: Inserting vertices of a full chunk into the global list of a vertex frontier with algorithm `graph500`.

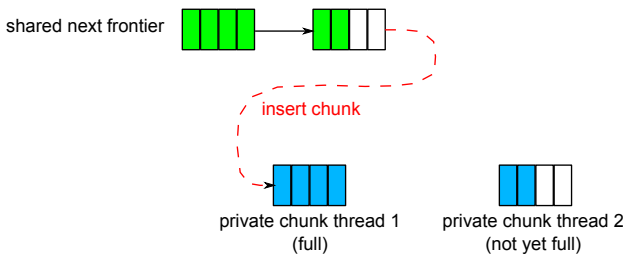


Fig. 4: Inserting vertices of a full chunk into the global list of a vertex frontier with algorithm `list`.

a logical ring topology [43] of the involved threads. Each thread keeps track of its neighbor's workload and supplies it with additional work, if it should be idle. A single control thread is present but the load balancing is done by the worker threads so the controller does not become a communication bottleneck. A downside of this optimization is that shifting work between threads may destroy data locality and affect overall performance.

Another approach to adapt the algorithm to the topology of the graph monitors the size of the next vertex frontier. At the end of an iteration, the number of active threads is adjusted to match the workload of the coming iteration [25]. Like the previous optimization this does not go well with data locality for NUMA architectures or distributed systems because vertices owned by an inactive process or thread would have to be moved to some active unit.

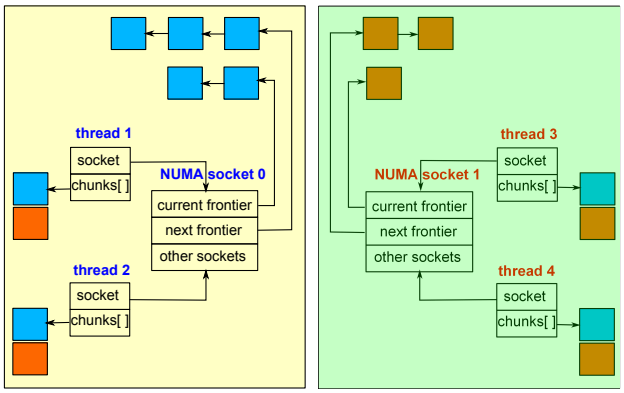


Fig. 5: Modeling a multicore NUMA architecture in software with algorithm `socketlist` (example for a 2-socket 2-core system).

III. Evaluated Algorithms

In our evaluation, we used the following parallel algorithms, each representing certain points in the described algorithm design space for shared memory systems, with an emphasis on level-synchronous algorithms:

- `global`: vertex-centric strategy as described in Section II-B, with parallel iterations over *all vertices on each level* [27]. The vertices are distributed statically in blocks to all threads. As pointed out already, this will only work for graphs with a very low depth. All data is allocated in parallel to meet the NUMA first touch policy and take care of data locality. First touch strategy [44] means that the processor that executes the initial access to a part of a data structure (usually with a granularity of a 4 KB page of the virtual address space) allocates that part of the data structure in the DRAM that is assigned to that NUMA node. This is the default allocation strategy in most operating systems including Linux. The distribution of data and work is therefore to split the vertices into blocks and assign each block to a different thread. Then, the data allocation as well as the work on that data is distributed.
- `graph500`: OpenMP reference implementation in the Graph500 benchmark [21] using a single array list with atomic Compare-And-Swap (CAS) and Fetch-And-Add accesses to insert chunks of vertices. Vertex insertion into core-/thread-local chunks is done without synchronized accesses. Only the insertion of a full chunk into the global list has to be done in a synchronized manner (atomically increasing a write index). All vertices of a full chunk get copied to the global array list. See Fig. 3 for a two thread example.
- `bag`: using OpenMP [45] tasks and two bag containers as described in [28]. This approach implicitly deploys load balancing mechanisms. Because of its parallel task based divide and conquer nature it does not take data locality into account (but leaves it solely to the thread runtime system). The original *bag* data structure is extended by a so called *hopper* [28]. This additional structure serves as a vertex cache to reduce the amount of insert operations on the main structure. In addition to the OpenMP implementation we implemented a Cilk+ version [46] as in the the original paper that did not show any significant differences to the OpenMP version with respect to performance.
- `list`: deploys two chunked linear lists with thread safe manipulators based on CAS operations. Threads concurrently remove chunks from the current node frontier and insert unvisited vertices into private chunks. Once a chunk is full, the chunk is inserted into the next node frontier, relaxing concurrent access. The main difference to `graph500` is that vertices are not copied to a global list but rather a whole chunk gets inserted (updating pointers only). And another difference to `graph500` is related to that the distribution of vertices of a frontier to

threads is chunk-based rather than vertex based. There is some additional overhead, if local chunks get filled only partially. See Fig. 4 for a two thread example.

- `socketlist`: extends the previous approach to respect data locality and NUMA awareness. The data is logically and physically distributed to all NUMA-nodes (i.e., processor sockets). Each thread primarily processes vertices of the current vertex frontier from its own NUMA-node list where the lists from the previous level are used for equal distribution of work. If a NUMA-node runs out of work, work is stolen from overloaded NUMA-nodes [17]. A newly detected vertex for the next vertex frontier is first inserted into a thread local chunk. Instead of using only one buffer chunk per NUMA node, *n* chunks are used per NUMA node by each thread where *n* is the number of NUMA nodes in the system. This simplifies NUMA-aware insertion of a chunk into the correct part of the next frontier. See Fig. 5 for a four threads in a 2-socket 2-core system.
- `bitmap`: further refinement and combination of the previous two approaches. A bitmap is used to keep track of visited vertices to reduce memory bandwidth. Again, built-in atomic CAS operations are used to synchronize concurrent access [17].

The first algorithm `global` is vertex-centric, all others are level-synchronous container-centric in our classification and utilize parallelism over the current vertex front. The last three implementations use a programming technique to trade (slightly more) redundant work against atomic operations as described in [20]. `socketlist` is the first container-centric algorithm in the list that pays attention to the NUMA memory hierarchy, `bitmap` additionally tries to reduce memory bandwidth by using an additional bitmap to keep track of the binary information whether a vertex is visited or not.

Some of the algorithms work with chunks were the chunk size is an algorithm parameter. For those algorithms we used a chunk size of 64 vertices. Another algorithm parameter in the threshold value in `bag`; we used 1,000 for that parameter. We did some pre experiments with these algorithm parameters using different graphs and different system and found that these chosen values were reasonable / best for most / many test instances.

IV. Experimental Setup

Beside the choice of the parallel algorithm itself certain parameters may influence the performance of parallel BFS, possibly dependent on the algorithm used. We are interested in relative performance comparisons between the different algorithms but also in their scalability for an increasing degree of parallelism. The latter aspect is of particular interest as future processors / systems will have more parallelism than todays' systems and that available hardware parallelism needs to be utilized efficiently by programs / algorithms. Therefore, scalability is a major concern. Large parallel systems with different architectures are used in the evaluation to examine the influence of the degree of parallelism and main system aspects.

Furthermore, the graph topology will likely have a significant influence on performance. Level-synchronous algorithms in general need on *each* level enough parallelism to feed all available threads. On the other side, if there is significantly more parallelism available than the number of threads, this parallelism has to be managed. Therefore, the available parallelism in each BFS level as well as the distribution of available parallelism over all levels is of interest when evaluating such algorithms.

In this section, we specify our parallel system test environment, describe classes of graphs and chosen graph representatives in this classes. The algorithms are implemented in C and partially C++ using the OpenMP parallel programming model [45]. Porting this to other parallel programming models utilizing thread-based shared memory parallelism like the new thread functionality in the recent language standards for C [47] and C++ [48], or using similar thread-based programming models like PThreads [49] or Cilk+ [46] should be rather straightforward.

To be able to handle also very large graphs, 64 bit indices were used throughout all tests unless otherwise stated. A discussion on using 32 bit indices (which can reduce memory bandwidth demands significantly) for graphs that are limited to roughly 4 billion vertices / edges is done in Section V-C.

A. Test Environment

Today, any mainstream system with more than one processor socket is organized as a NUMA system. Fig. 1 has shown the principal system architecture of such a system, in the example shown for a 4 socket system. On such systems, especially data-intensive algorithms have to respect distributed data allocation [44] [50] and processor locality [19] in the execution of an algorithm. Beside the performance characteristics of the memory hierarchy known from modern processors [7], in a NUMA system an additional penalty exists if a core accesses data that is not cached by a private L1/L2 cache of this core or the shared L3 cache of the corresponding processor, and the data resides in that part of the distributed main memory that is allocated on a different NUMA socket.

We used in our tests parallel systems (see Table I for details) that span a spectrum of system parameters, mainly the degree of parallelism, NUMA topology, cache sizes, and cycle time. The largest system is a 64-way AMD-6272 Interlagos based system with 128 GB shared memory organized in 4 NUMA nodes, each with 16 cores. An additional AMD based system with 4 NUMA nodes but fewer core count was used, too. Two other systems are Intel based systems with only 2 NUMA nodes each. We will focus our discussion on the larger Interlagos system and discuss in Section V-C the influence of the system details.

B. Graphs

It is obvious that graph topology will have a significant influence on the performance of parallel BFS algorithms. We

TABLE I: SYSTEMS USED.

name	Intel-IB	Intel-SB	AMD-IL	AMD-MC
processor:				
manufacturer	Intel	Intel	AMD	AMD
CPU nodel	E5-2697	E5-2670	Opteron 6272	Opteron 6168
architecture	Ivy Bridge	Sandy Bridge	Interlagos	Magny Cours
frequency[GHz]	2.7	2.6	2.1	1.9
last level cache size [MB]	30	20	16	12
system:				
memory [GB]	256	128	128	32
number of CPU sockets	2	2	4	4
n-way parallel	48	32	64	48

used some larger real graphs from the DIMACS-10 challenge [51], the Florida Sparse Matrix Collection [52], and the Stanford Large Dataset Collection [53]. Additionally, we used synthetically generated pseudo-random graphs that guarantee certain topological properties. R-MAT [54] is such a graph generator with parameters a, b, c influencing the topology and clustering properties of the generated graph (see [54] for details). R-MAT graphs are mostly used to model scale-free graphs. The graph `friendster` and larger RMAT-graphs could not be used on all systems due to memory requirements. We used in our tests graphs of the following classes:

- Graphs with a very low average and maximum vertex degree resulting in a rather high graph depth and limited vertex fronts. A representative for this class is the road network `road-europe`.
- Graphs with a moderate average and maximum vertex degree. For this class we used Delaunay graphs representing Delaunay triangulations of random points (`delaynay`) and a graph for a 3D PDE-constraint optimization problem (`nlpkkt240`).
- Graphs with a large variation of degrees including few very large vertex degrees. Related to the graph size, they have a smaller graph depth. For this class of graphs we used a real social network (`friendster`), link information for web pages (`wikipedia`), and synthetically generated Kronecker R-MAT graphs with different vertex and edge counts and three R-MAT parameter sets. The first parameter set named 30 is $a = 0.3, b = 0.25, c = 0.25$, the second parameter set 45 is $a = 0.45, b = 0.25, c = 0.15$, and the third parameter set 57 is $a = 0.57, b = 0.19, c = 0.19$.

All our test graphs are connected, for R-MAT graphs guaranteed with $n-1$ artificial edges connecting vertex i with vertex $i+1$. Some important graph properties are given in Table II. For a general discussion on degree distributions of R-MAT graphs see [55].

V. RESULTS

In this section, we discuss our results for the described test environment. Performance results will be given in *Million Traversed Edges Per Second* $MTEPS := m/t/10^6$, where m is the number of edges and t is the elapsed time in seconds an algorithm takes. MTEPS is a common metric for BFS performance [21] (higher is better). To give an idea on the elapsed time, for example an MTEPS value of 2000 for the graph `RMAT-1M-1G-57` with 1 million vertices and 1 billion edges corresponds to 250 milliseconds execution time for the whole graph traversal on the system Intel-SB. In an undirected graph representing an edge internally with two edges (u,v) and (v,u) only half of the internal edges are counted in this metric.

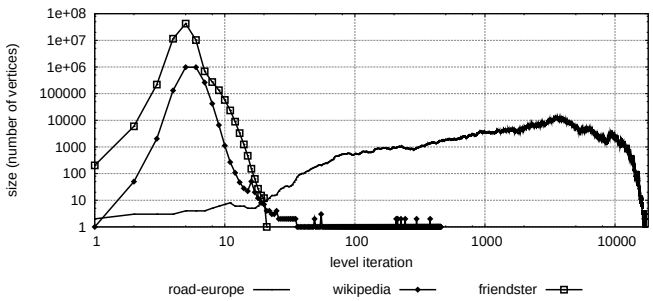


Fig. 6: Dynamic sizes of some vertex frontiers and potential parallelism.

On large and more dense graphs, MTEPS values are generally higher than on very sparse graphs. This is due to the fact that in denser graphs many visited edges do not generate an *additional* entry (and therefore work) in a container of unvisited vertices. This observation is not true for the algorithm `global`, where in all levels all vertices get traversed. The MTEPS numbers for the graphs and systems used vary between less than 1 and approx. 3,500, depending on the graph, system and algorithm.

In the following discussion on results, we distinguish between different views on the problem. It is not possible to show all our results in this paper in detail (4 parallel systems, 26 different graphs, up to 11 thread counts, 32/64 bit versions, different compilers / compiler switches). Rather than that, we summarize results and show only interesting or representative aspects in detail.

A. Graph Properties and Scalability

In terms of scalability, parallel algorithms need enough parallel work to feed all threads. For graphs with limiting properties, such as very small vertex degrees or small total number of vertices / edges, there are problems to feed many parallel threads. Additionally, congestion in accessing smaller shared data structures arise.

Fig. 6 shows relevant vertex frontier sizes for 3 selected graphs showing different characteristics. The x axis gives the level of BFS traversal, the y axis shows the corresponding

TABLE II: CHARACTERISTICS OF THE USED GRAPHS.

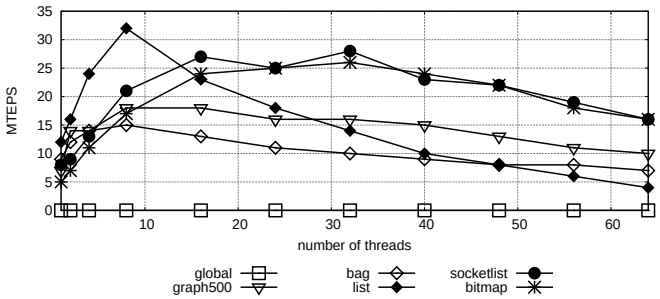
graph name	$\|V\| \times 10^6$	$\|E\| \times 10^6$	degree avg.	degree max.	graph depth
delaunay (from [51])	16.7	100.6	6	26	1650
nlpkkt240 (from [52])	27.9	802.4	28.6	29	242
road-europe (from [51])	50.9	108.1	2.1	13	17345
wikipedia (from [52])	3.5	45	12.6	7061	459
friendster (from [53])	65.6	3612	55	5214	22
RMAT-1M-10M-30	1	10	10	107	11
RMAT-1M-10M-45	1	10	10	4726	16
RMAT-1M-10M-57	1	10	10	43178	400
RMAT-1M-100M-30	1	100	100	1390	9
RMAT-1M-100M-45	1	100	100	58797	8
RMAT-1M-100M-57	1	100	100	530504	91
RMAT-1M-1G-30	1	1000	1000	13959	8
RMAT-1M-1G-45	1	1000	1000	599399	8
RMAT-1M-1G-57	1	1000	1000	5406970	27
RMAT-100M-1G-30	100	1000	10	181	19
RMAT-100M-1G-45	100	1000	10	37953	41
RMAT-100M-1G-57	100	1000	10	636217	3328
RMAT-100M-2G-30	100	2000	20	418	16
RMAT-100M-2G-45	100	2000	20	85894	31
RMAT-100M-2G-57	100	2000	20	1431295	1932
RMAT-100M-4G-30	100	4000	40	894	15
RMAT-100M-4G-45	100	4000	40	180694	31
RMAT-100M-4G-57	100	4000	40	3024348	1506
RMAT-100M-8G-30	100	8000	40	1807	15
RMAT-100M-8G-45	100	8000	40	371454	21
RMAT-100M-8G-57	100	8000	40	6210095	1506

size of the vertex frontier for this level. The frontier size for `friendster` has a steep curve (i.e., there is soon enough parallelism available) that remains high for nearly all level iterations. The frontier size for `wikipedia` start similar, but has for the later level iterations only few vertices per frontier left, which restricts *any* level-synchronous BFS algorithm in utilizing parallelism in this later level iterations. And the worst case shown is the graph `road-europe` where the frontier size never exceeds more than roughly 10,000 vertices. Working on 10,000 vertices with 64 threads means that every thread has not more than roughly 150 vertices to work on, and the computational density for BFS is rather low.

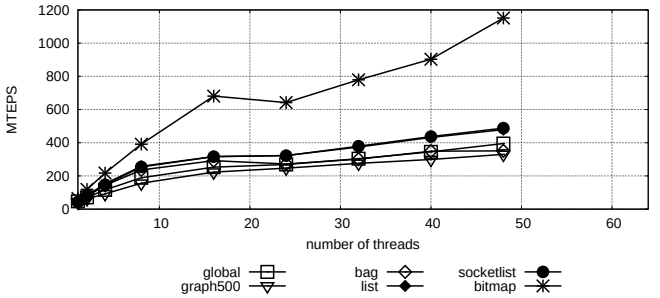
For graphs with such limiting properties (road network, the delaunay graph and partially small RMAT-graphs), for *all* analyzed algorithms performance is limited or even drops as soon the number of threads is beyond some threshold; on *all* of our systems around 8-16 threads. Fig. 7a shows the worst case of such an behavior with `road-europe`. For graphs with such properties, other algorithms different to a level-synchronous approach should be taken into account, e.g., [16].

For large graphs and / or high vertex degrees (all larger R-MAT graphs, `friendster`, `nlpkkt240`), the results were quite different from that and all algorithms other than `global` showed on nearly all such graphs and with few exceptions a continuous but in detail different performance increase over all thread counts (see Fig. 7b for an example and the detailed discussion below). Best speedups reach nearly 40 (`bitmap` with `RMAT-1M-1G-30`) on the 64-way parallel system.

For denser graphs with a very low depth often all algorithms show a very similar behavior and even absolute performance, even with an increasing number of threads. See Fig. 8 for an example.



(a) Limited scalability with graph `road-europe` on system AMD-IL.



(b) Continuous performance increase with more threads for graph `friendster` on system Intel-IB.

Fig. 7: Differences in Scalability.

B. Algorithms

For small thread counts up to 4-8, all algorithms other than `global` show with few exceptions and within a factor of

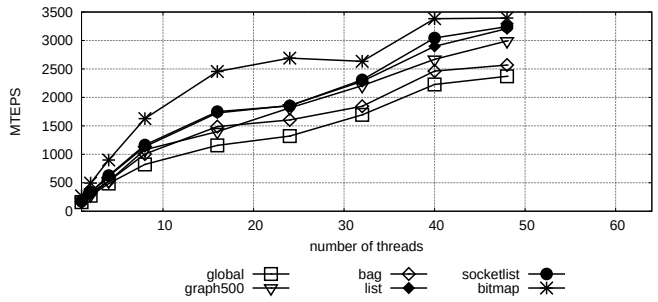
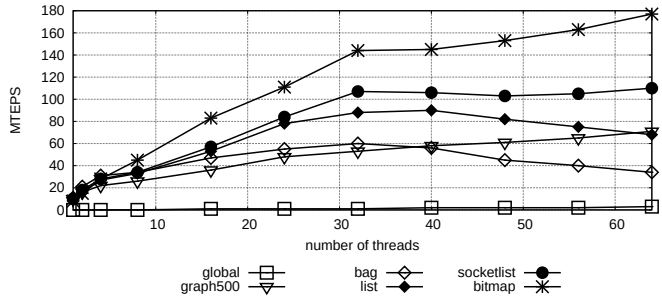
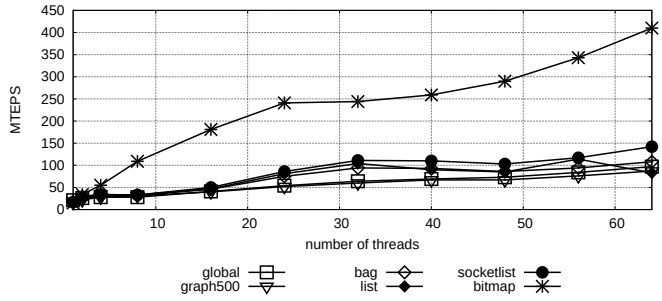


Fig. 8: Similar principal behavior for dense graphs with a small depth like on the graph RMAT-1M-1G-30 on system Intel-IB with 32 bit indices.



(a) Benefit of NUMA awareness for graph RMAT-100M-1G-57 on system AMD-IL.



(b) Memory bandwidth optimization with algorithm bitmap for graph friendster on system AMD-IL.

Fig. 9: Benefits of clever memory handling.

2 comparable results in absolute performance and principal behavior. Therefore, for small systems / a low degree of parallelism the choice of algorithm is not really crucial. But for large thread counts, algorithm behavior can be quite different. Therefore, we concentrate the following discussion on individual algorithms primarily on large thread counts (more than 8).

The algorithm global has a very different approach than all other algorithms, which can be also easily seen in the results. For large graphs with low vertex degrees, this algorithm performs extremely poor as many level-iterations are necessary, e.g., factor 100 slower for the road graph compared to the second worst algorithm; see Fig. 7a for an example of that behavior. The algorithm is only competitive on the systems we used if the graph is very small and the graph depth is very low resulting in only a few level-iterations, e.g., less than 10. See Fig. 8 for an example.

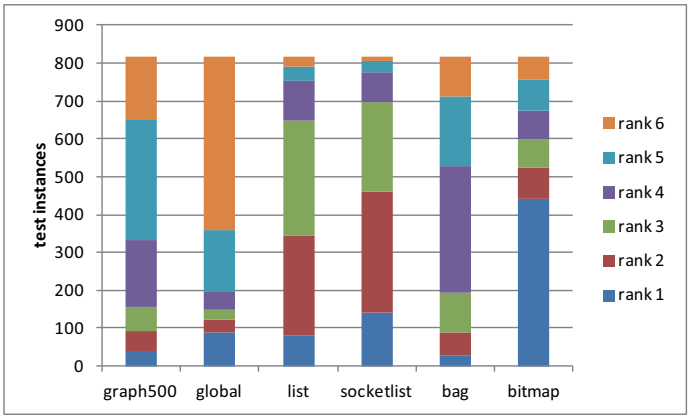
The graph500 algorithm uses atomic operations to increment the position where (a chunk of) vertices get to be inserted into the new vertex front. Additionally, all vertices of a local chunk get copied to the global list (vertex front). This can be fast as long as the number of processors is small. But, as the number of threads increases, the cost *per atomic operation* increases [20], and therefore, the performance drops often significantly relative to other algorithms. Additionally, this algorithm does not respect data / NUMA locality on copying vertices from a local chunk to a global list, which gets a serious problem with large thread counts.

Algorithm bag shows only good results for small thread counts or dense graphs. Similar to graph500, this algorithm is not locality / NUMA aware. The bag data structure is based on smaller substructures. Because of the recursive and task parallel nature of the algorithm, the connection between the allocating thread and the data is lost, often destroying data locality as the thread count increases. Respecting locality is delegated solely to the run-time system mapping tasks to cores / NUMA nodes. In principle, it is often a good idea to delegate complex decisions to runtime systems. But in this case the runtime system (without modifications / extensions) has not enough knowledge about the whole context that would be necessary to further optimize the affinity handling in a more global view. Explicit affinity constructs as in the latest OpenMP version 4.0 [45] could be interesting for that to explicitly optimize this algorithm for sparser graphs or many threads instead of leaving all decisions to the runtime system.

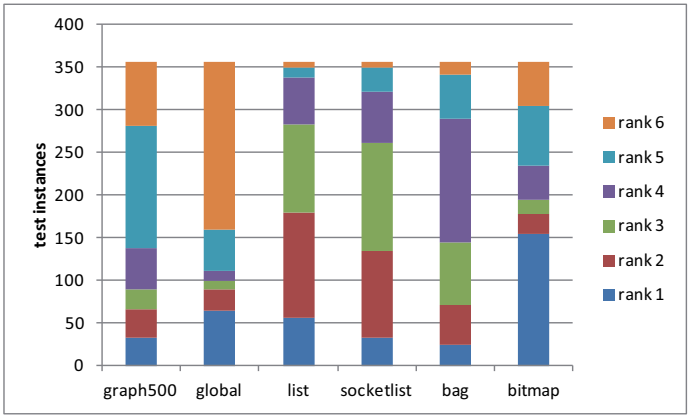
The simple list algorithm has good performance values for small thread counts. But for many threads, list performs rather poor on graphs with high vertex degrees. Reasons are implementation specific the use of atomic operations for insert / remove of full / final chunks and that in such vertex lists processor locality is not properly respected. When a thread allocates memory for a vertex chunk and inserts this chunk into the next node frontier, the chunk might be dequeued by another thread in the next level iteration. This thread might be executed on a different NUMA-node, which results in remote memory accesses. This problem becomes larger with increasing thread / processor counts. The list algorithm has on the other side a very low computational overhead such that this algorithms is often very good with a small thread count.

The socketlist approach improves the list idea with respect to NUMA aware data locality at the expensive of an additional more complex data structure. For small thread counts, this is an additional overhead that often does not pay off but on the other side also does not drop performance in a relevant way. But for larger thread counts, the advantage is obvious looking at the cache miss and remote access penalty time of current and future processors (see Fig. 9).

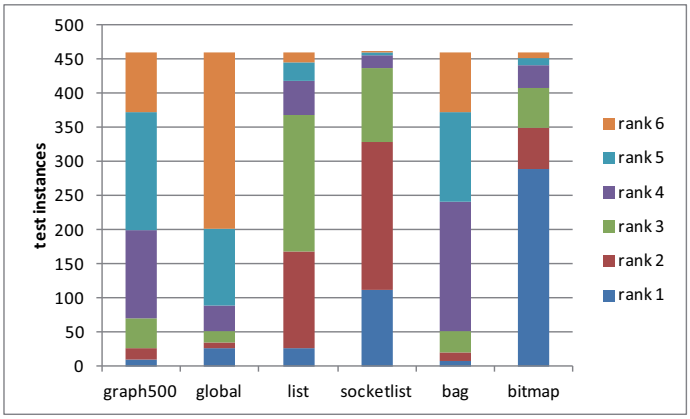
The additional overhead of the bitmap algorithm makes this algorithm with only a few threads even somewhat slower than some other algorithms (but again not in a relevant way). But the real advantage shows off with very large and dense graphs and large thread counts, when even higher level caches are not



(a) Algorithm ranking for all test instances.



(b) Algorithm ranking for a thread count up to 8.



(c) Algorithms ranking for a thread count 16 and more.

Fig. 10: Relative performance ranking of algorithms.

sufficient to buffer vertex fronts and memory bandwidth gets the real bottleneck. The performance difference to all other algorithms can then be significant and is even higher with denser graphs (see Figs. 8, 9a, 9b).

In the above discussion performance observations on algorithms were given in a general way but explicitly shown only for selected examples. Fig. 10 shows summarizing statistics on the algorithms for all graphs on all systems and all thread counts. The six algorithms of interest are ranked for each problem instance (graph, system, thread count) on their relative performance to each other with a rank from 1 to 6. An algorithm ranked first for a problem instance was the best performing algorithm for that problem instance. A rank says nothing about the absolute difference between two algorithms for a problem instance. The difference between two performance numbers might be rather small while for another problem instance the performance of the algorithm ranked first might be significant higher then the performance of the second ranked algorithm.

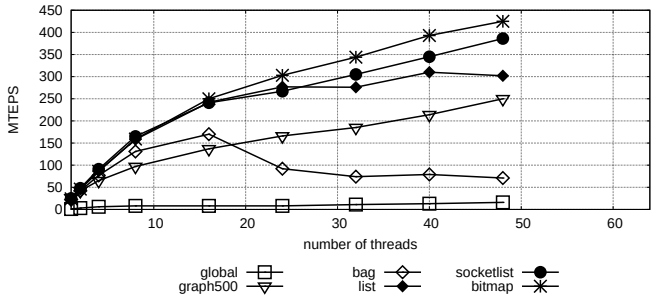
Fig. 10 shows three histograms: for all test instances, for thread counts up to 8 (i.e., small parallel systems), and for threads counts of 16 and more (i.e., large parallel systems). As a remark, the results for very different graphs topologies are summarized in these diagrams such that for example for very small graphs algorithms with a startup overhead to generate complex data structures have a disadvantage and may be ranked lower. As can be clearly seen, the algorithms that optimize memory accesses (awareness of NUMA topology, memory bandwidth reduction) show best results. This is especially true for many threads and for the two systems with 4 NUMA nodes (see for example Figs. 7b and 9b for this observation). The algorithm `bitmap` was ranked first or second in more than 75% of all problems instances with large threads counts and roughly 50% even for small threads counts. Also, evidently the algorithm `global` is worst in more than halve of all instances.

C. Influence of the system architecture

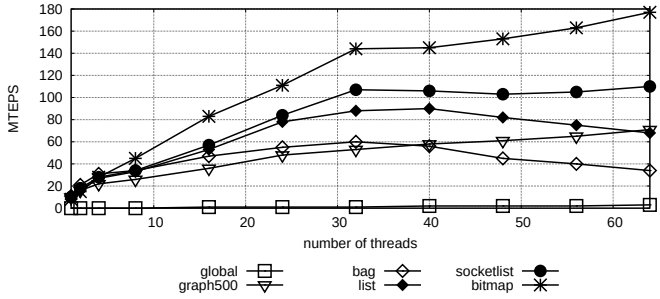
As described in Section IV, we used in our tests different systems but concentrated our discussions so far on results on the largest AMD-IL system. While the principle system architecture on Intel and AMD systems got in the last years rather similar, implementation details, e.g., on cache coherence, atomic operations, cache sizes, and the micro-architecture are quite different [56] [57].

While the Intel systems are 2 socket systems, the AMD systems are 4 socket system, and the latter systems showed (as expected) more sensibility to locality / NUMA. Fig. 11 shows this sensibility for the same graph on a 4 socket system and on a 2 socket system. While on the 4 NUMA node system the `bitmap` algorithm has more advantages than all other algorithms, on the less sensitive 2 NUMA node system the performance difference of the algorithms to all other algorithms is less.

Hyper-Threading on Intel systems gave improvements only for large RMAT graphs. There is a choice of using 32 bit indices (i.e., the data type **`int`** or **`unsigned int`**) or 64 bit indices (i.e., the data type **`long`** or **`unsigned long`**). Using a 32 bit index limits the number of vertices / edges to not more than roughly 4 billion. On the other side, a 32 bit index requires less memory bandwidth, which is rather preciously for a BFS algorithm with very low computational density. Comparing 32 bit ro 64 bit results showed as expected performance improvements due to lower memory bandwidth requirements

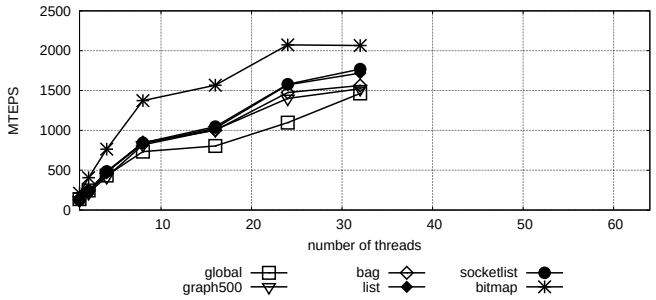


(a) 2 NUMA nodes for graph RMAT-100M-1G-57 on system Intel-IB.

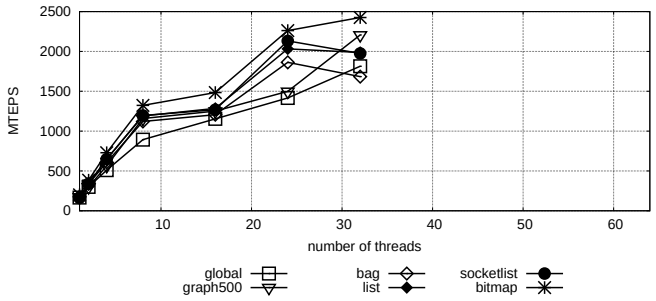


(b) 4 NUMA nodes for graph RMAT-100M-1G-57 on system AMD-IL.

Fig. 11: Difference in NUMA nodes.



(a) 64 bit indices for graph RMAT-1M-1G-30 on system Intel-SB.



(b) 32 bit indices for graph RMAT-1M-1G-30 on system Intel-SB.

Fig. 12: Switching from 64 to 32 bit indices.

and fitting more information in caches. These improvements were around 20-30% for all algorithms other than bitmap (with a less bandwidth pressure than the other algorithms). Fig. 12 shows an example for this improvement. The absolute improvement is for example for 32 threads roughly 1800 MTEPS with 32 bit indices compared to roughly 1500 MTEPS with 64 bit indices. The advantage of the algorithm bitmap compared to all other algorithms decreases as the pressure on memory bandwidth in decreased with 32 bit instead of 64 bit indices.

VI. Conclusions

In our evaluation for a selection of parallel level synchronous BFS algorithms for shared memory systems, we showed that for small systems / a limited number of threads all algorithms other than global behaved almost always rather similar, including absolute performance. Therefore, for very small systems the choice of parallel BFS algorithm other than global is not crucial.

But using large parallel NUMA-systems with a deep memory hierarchy, the evaluated algorithms show often significant differences. Here, the NUMA-aware algorithms socketlist and bitmap showed constantly good performance and good scalability, if vertex fronts are large enough. Both algorithms utilize dynamic load balancing combined with locality handling, this combination is a necessity on larger NUMA systems. Especially on larger and more dense graphs, the bitmap shows often significant performance advantages over all other algorithms (approx. 75% of all test instances ranked first or second). Using 32 bit indices for smaller graphs instead of 64 bit indices reduces the benefit of this algorithm compared to the others.

The level-synchronous approach should be used only if the graph topology ensures enough parallelism on each / on most levels for the given system. Otherwise, *any* level-synchronous BFS algorithm has problems to feed that many threads.

Acknowledgements

The system infrastructure was partially funded by an infrastructure grant of the Ministry for Innovation, Science, Research, and Technology of the state North-Rhine-Westphalia.

We thank the anonymous reviewers for their careful reading and their helpful comments and suggestions on an earlier version of the manuscript.

References

[1] R. Berrendorf and M. Makulla, "Level-synchronous parallel breadth-first search algorithms for multicore- and multiprocessors systems," in *Proc. Sixth Intl. Conference on Future Computational Technologies and Applications (FUTURE COMPUTING 2014)*, 2014, pp. 26–31.

[2] R. Sedgewick, *Algorithms in C++, Part 5: Graph Algorithms*, 3rd ed. Addison-Wesley Professional, 2001.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. The MIT Press, 2009.

[4] C. Wilson, B. Boe, A. Sala, K. P. Puttaswamy, and B. Y. Zhao, "User interactions in social networks and their implications," in *Proc. 4th ACM European Conference on Computer Systems (Eurosys)*, 2009, pp. 205–218.

[5] J. G. Siek, L.-Q. Lee, and A. Lumbdsdaine, *The Boost Graph Library*. Addison-Wesley Professional, 2001.

[6] P. S. Pacheco, *An Introduction to Parallel Programming*. Burlington, MA: Morgan Kaufman, 2011.

[7] J. L. Hennessy and D. A. Patterson, *Computer Architecture: A Quantitative Approach*, 5th ed. Morgan Kaufmann Publishers, Inc., 2012.

[8] Intel, *Intel® Quickpath Interconnect Maximizes Multi-Core Performance*, http://www.intel.com/technology/quickpath/, retrieved: 22.11.2014.

[9] Hypertransport Consortium, http://www.hypertransport.org/, retrieved: 22.11.2014.

[10] A. Yoo, E. Chow, K. Henderson, W. McLendon, B. Hendrickson, and U. Catalyurek, "A scalable distributed parallel breadth-first search algorithm on BlueGene/L," in *ACM/IEEE Conf. on Supercomputing*, 2005, pp. 25–44.

[11] F. Checconi, J. Willcock, and A. R. Choudhury, "Breaking the speed and scalability barriers for graph exploration on distributed-memory machines," in *Proc. Intl. Conference on High-Performance Computing, Networking and Storage and Analysis (SC'12)*, 2012, pp. 1–12.

[12] Message Passing Interface Forum, "MPI: A message-passing interface standard, version 3.0," Tech. Rep., Jul. 2012.

[13] S. Hong, S. Kim, T. Oguntebi, and K. Olukotun, "Accelerating CUDA graph algorithms at maximum warp," in *Proc. 16th ACM Symp. Principles and Practice of Parallel Processing (PPoPP)*, 2011, pp. 267–276.

[14] D. Li and M. Becchi, "Deploying graph algorithms on GPUs: an adaptive solution," in *Proc. 27nd Intl. Symp. on Parallelism and Distributed Computing (IPDPD2013)*. IEEE, 2013, pp. 1013–1024.

[15] *CUDA Toolkit Documentation v6.0*, Nvidia, http://docs.nvidia.com/cuda/, 2014, retrieved: 22.11.2014.

[16] J. D. Ullman and M. Yannakakis, "High-probability parallel transitive closure algorithms," *SIAM Journal Computing*, vol. 20, no. 1, pp. 100–125, 1991.

[17] V. Agarwal, F. Petrini, D. Pasetto, and D. A. Bader, "Scalable graph exploration on multicore processors," in *ACM/IEEE Intl.Conf. for High Performance Computing, Networking, Storage and Analysis (HPCNSA)*, 2010, pp. 1–11.

[18] J. Chhungani, N. Satish, C. Kim, J. Sewall, and P. Dubey, "Fast and efficient graph traversal algorithm for CPUs: Maximizing single-node efficiency," in *Proc. 26th Intl. Parallel and Distributed Processing Symposium*. IEEE, 2012, pp. 378–389.

[19] A. Agarwal and A. Gupta, "Temporal, processor and spatial locality in multiprocessor memory references," *Frontiers of Computing Systems Research*, vol. 1, pp. 271–295, 1990.

[20] R. Berrendorf, "Trading redundant work against atomic operations on large shared memory parallel systems," in *Proc. Seventh Intl. Conference on Advanced Engineering Computing and Applications in Sciences (ADVCOMP)*, 2013, pp. 61–66.

[21] Graph 500 Comitee, *Graph 500 Benchmark Suite*, http://www.graph500.org/, retrieved: 22.11.2014.

[22] D. Bader and K. Madduri, "SNAP, small-world network analysis and partitioning: an open-source parallel graph framework for the exploration of large-scale networks," in *22nd IEEE Intl. Symp. on Parallel and Distributed Processing*, 2008, pp. 1–12.

[23] N. Edmonds, J. Willcock, A. Lumsdaine, and T. Hoefler, "Design of a large-scale hybrid-parallel graph library," in *Intl. Conference on High Performance Computing Student Reserarch Symposium*. IEEE, Dec. 2010.

[24] J. Shun and G. E. Blelloch, "Ligra: A lightweight graph processing framework for shared memory," in *Proc. Principles and Practice of Parallel Processing*. IEEE, 2013, pp. 135–146.

[25] Y. Xia and V. Prasanna, "Topologically adaptive parallel breadth-first search on multicore processors," in *21st Intl. Conf. on Parallel and Distributed Computing and Systems*, 2009, pp. 1–8.

[26] D. Bader and K. Madduri, "Designing multithreaded algorithms for breadth-first search and st-connectivity on the Cray MTA-2," in *35th Intl. Conf. on Parallel Processing*, 2006, pp. 523–530.

[27] S. Hong, T. Oguntebi, and K. Olukotun, "Efficient parallel graph exploration on multi-core CPU and GPU," in *Intl. Conf. on Parallel Architectures and Compilation Techniques*, 2011, pp. 78–88.

[28] C. E. Leiserson and T. B. Schardl, "A work-efficient parallel breadth-first search algorithm (or how to cope with the nondeterminism of reducers)," in *Proc. 22nd ACM Symp. on Parallelism in Algorithms and Architectures*, 2010, pp. 303–314.

[29] T. B. Schardl, "Design and analysis of a nondeterministic parallel breadth-first search algorithm," Master's thesis, MIT, EECS Department, Jun. 2010.

[30] P. Harish and P. Narayanan, "Accelerating large graph algorithms on the GPU using CUDA," in *14th Intl. Conf. on High Performance Computing*, 2007, pp. 197–208.

[31] P. Harish, V. Vineet, and P. Narayanan, "Large graph algorithms for massively multithreaded architectures," IIIT Hyderabad, Tech. Rep., 2009.

[32] L. Luo, M. Wong, and W. Hwu, "An effective GPU implementation of breadth-first search," in *47th Design Automation Conference*, 2010, pp. 52–55.

[33] S. Beamer, K. Asanovic, and D. Patterson, "Direction-optimizing breadth-first search," in *Proc. Supercomputing 2012*, 2012, pp. 1–10.

[34] Y. Yasui, K. Fujusawa, and K. Goto, "NUMA-optimized parallel breadth-first search on multicore single-node system," in *Proc. IEEE Intl. Conference on Big Data*, 2013, pp. 394–402.

[35] D. Merrill, M. Garland, and A. Grimshaw, "Scalable GPU graph traversal," in *Proc. Principles and Practice of Parallel Processing*. IEEE, 2012, pp. 117–127.

[36] L.-M. Munguìa, D. A. Bader, and E. Ayguade, "Task-based parallel breadth-first search in heterogeneous environments," in *Proc. Intl. Conf. on High Performance Computing (HiPC 2012)*, 2012, pp. 1–10.

[37] A. Buluç and K. Madduri, "Parallel breadth-first search on distributed memory systems," in *Proc. Supercomputing*. IEEE, 2011, pp. 65–79.

[38] H. Gazit and G. L. Miller, "An improved parallel algorithm that computer the BFS numbering of a directed graph," *Information Processing Letters*, vol. 28, pp. 61–65, Jun. 1988.

[39] R. K. Gosh and G. Bhattacharjee, "Parallel breadth-first search algorithms for trees and graphs," *Intern. Journal Computer Math.*, vol. 15, pp. 255–268, 1984.

[40] H. Lv, G. Tan, M. Chen, and N. Sun, "Understanding parallelism in graph traversal on multi-core clusters," *Computer Science – Research and Development*, vol. 28, no. 2-3, pp. 193–201, 2013.

[41] D. Scarpazza, O. Villa, and F. Petrini, "Efficient breadth-first search on the Cell/BE processor," *IEEE Trans. Par. and Distr. Systems*, pp. 1381–1395, 2008.

[42] R. Pearce, M. Gokhale, and N. M. Amato, "Scaling techniques for massive scale-free graphs in distributed (external) memory," in *Proc. Intl. Symposium on Parallel and Distributed Processing (IPDPS)*. IEEE, 2013, pp. 825–836.

[43] Y. Zhang and E. Hansen, "Parallel breadth-first heuristic search on a shared-memory architectur," in *AAAI Workshop on Heuristic Search, Memory-Based Heuristics and Their Applications*, 2006, pp. 1 – 6.

[44] B. Chapman, G. Jost, and R. van der Pas, *Using OpenMP*. Cambridge, MA: The MIT Press, 2008.

[45] *OpenMP Application Program Interface*, 4th ed., OpenMP Architecture Review Board, http://www.openmp.org/, Jul. 2013, retrieved: 22.11.2014.

[46] *Intel® Cilk™Plus*, https://software.intel.com/en-us/intel-cilk-plus, retrieved: 22.11.2014.

[47] *ISO/IEC 9899:2011 - Programming Languages – C*, ISO, Genf, Schweiz, 2011.

[48] *ISO/IEC 14882:2011 Programming Languages – C++*, ISO, Genf, Schweiz, 2011.

[49] IEEE, *Posix.1c (IEEE Std 1003.1c-2013)*, Institute of Electrical and Electronics Engineers, Inc., 2013.

[50] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon, *Parallel Programming in OpenMP*. San Francisco: Morgan Kaufman Publishers, 2001.

[51] DIMACS, *DIMACS'10 Graph Collection*, http://www.cc.gatech.edu/dimacs10/, retrieved: 22.11.2014.

[52] T. Davis and Y. Hu, *Florida Sparse Matrix Collection*, http://www.cise.ufl.edu/research/sparse/matrices/, retrieved: 22.11.2014.

[53] J. Leskovec, *Stanford Large Network Dataset Collection*, http://snap.stanford.edu/data/index.html, retrieved: 22.11.2014.

[54] D. Chakrabarti, Y. Zhan, and C. Faloutsos, "R-MAT: A recursive model for graph mining," in *SIAM International Conference on Data Mining*, 2004, pp. 442 – 446.

[55] C. Groër, B. D. Sullivan, and S. Poole, "A mathematical analysis of the R-MAT random graph generator," *Networks*, vol. 58, no. 3, pp. 159–170, Oct. 2011.

[56] Intel, *Intel® 64 and IA-32 Architectures Optimization Reference Manual*, 2014.

[57] Advanced Micro Devices, *Software Optimization Guide for AMD Family 15h Processors*, Jan. 2012.

www.iariajournals.org

International Journal On Advances in Intelligent Systems
issn: 1942-2679

International Journal On Advances in Internet Technology
issn: 1942-2652

International Journal On Advances in Life Sciences
issn: 1942-2660

International Journal On Advances in Networks and Services
issn: 1942-2644

International Journal On Advances in Security
issn: 1942-2636

International Journal On Advances in Software
issn: 1942-2628

International Journal On Advances in Systems and Measurements
issn: 1942-261x

International Journal On Advances in Telecommunications
issn: 1942-2601